

INTUITIONISTIC IMPLICATION MAKES MODEL CHECKING HARD*

MARTIN MUNDHENK AND FELIX WEISS

Universität Jena, Institut für Informatik, Jena, Germany
e-mail address: {martin.mundhenk,felix.weiss}@uni-jena.de

ABSTRACT. We investigate the complexity of the model checking problem for intuitionistic and modal propositional logics over transitive Kripke models. More specific, we consider intuitionistic logic IPC, basic propositional logic BPL, formal propositional logic FPL, and Jankov's logic KC. We show that the model checking problem is P-complete for the implicational fragments of all these intuitionistic logics. For BPL and FPL we reach P-hardness even on the implicational fragment with only one variable. The same hardness results are obtained for the strictly implicational fragments of their modal companions. Moreover, we investigate whether formulas with less variables and additional connectives make model checking easier. Whereas for variable free formulas outside of the implicational fragment, FPL model checking is shown to be in LOGCFL, the problem remains P-complete for BPL.

1. INTRODUCTION

Intuitionistic propositional logic IPC (see e.g. [31]) goes back to Heyting and bases on Brouwer's idea of constructivism from the beginning of the 20th century. It can be seen as the part of classical propositional logic that goes without the use of the excluded middle $a \vee \neg a$.

While it was originally conceived and is primarily of interest from a proof-theoretic point of view, IPC admits many sound and complete semantics, such as the algebraic semantics [29], the topological semantics [17], and the arithmetical semantics [7]. The most well known semantics for IPC is Kripke's possible world semantics [15]. As a matter of fact, already in the 1930s it was observed by Gödel that IPC can be mapped to a fragment of the modal logic S4, which was later shown to be the modal logic of the class of transitive and reflexive Kripke frames [14]. In this paper, we explore this Kripke semantics further.

Whereas the complexity of the validity problem for IPC is deeply studied [25, 28, 22, 12], the exact complexity of its model checking problem is open. Research on the complexity of model checking on Kripke models goes back to [10, 23] (where it is called determination of truth) and has been done for a variety of logics like dynamic logic and many temporal logics. It was recently shown that the model checking problem for IPC formulas with one variable is AC¹-complete [19]. We investigate the complexity of model checking for different

1998 ACM Subject Classification: F.2, F.4.

Key words and phrases: complexity, intuitionistic logic, model checking, P-completeness.

* Preliminary parts of the results appeared in extended abstracts [18, 19].

intuitionistic logics and for related modal logics—their modal companions. Our central question is which ingredients (i.e. logical connectives, number of variables) are needed in order to obtain maximal hardness of the model checking problem.

We consider the intuitionistic logics BPL (basic propositional logic [32]), FPL (formal propositional logic [32]), IPC and KC (Jankov’s logic, see [9]). All have semantics that is defined over Kripke models with a monotone valuation function and a transitive frame¹ (as for BPL) that distinguish on whether the frame is additionally irreflexive (FPL), reflexive (IPC), or a directed preorder (KC). The validity problem for all these logics is PSPACE-complete [25, 4, 28], and the satisfiability problem is NP-complete for IPC and for KC, but in NC¹ for BPL and for FPL. These intuitionistic logics can be embedded into the modal logics K4, PrL (provability logic [1, 26]), S4, and S4.2, that are called the modal companions of the respective intuitionistic logic. The validity problem and the satisfiability problem is PSPACE-complete for all these modal logics [16, 24]. The PSPACE-completeness results mentioned also hold for the implicational fragment of intuitionistic logics [25, 4, 28] resp. the strictly implicational fragment for the considered modal logics [2]. Also, the complexity of the validity problem for fragments of the considered logics with a bounded number of variables was investigated [27, 5, 22]. Roughly speaking, the number of variables that is needed to obtain a PSPACE-hard validity problem depends on whether the semantics restricts the transitive frames (of the Kripke models) to be reflexive, irreflexive, or none of both. For intuitionistic logics, it is shown in [22] that on transitive and reflexive frames (IPC) one needs two variables to reach PSPACE-hardness for the validity problem, on transitive and irreflexive frames (FPL) one variable is necessary, and on arbitrary transitive frames (BPL) one comes out without variables at all. For their modal companions, the same bounds apply for transitive and irreflexive frames (PrL) [27] and for arbitrary transitive frames (K4) [5], but for transitive and reflexive frames (S4) already one variable suffices [5]. Notice that no PSPACE-hardness results are known for the implicational fragment with a bounded number of variables.

The model checking problem is the following decision problem. Given a formula, a Kripke model, and a state in this model, decide whether the formula is satisfied in that state. For classical propositional logic, the model checking problem (also called the formula evaluation problem) can be solved in alternating logarithmic time [3]. Since the models for classical propositional logic can be seen as a special case of Kripke models that consist of only one state, we cannot expect such a low complexity for intuitionistic logics, where the models may consist of many states. For the considered logics, the upper bound P follows from [10]. In fact, this upper bound turns out to be the lower bound too—we show that the model checking problem for KC, IPC, BPL, and FPL is P-complete, even on the implicational fragments. We obtain the same bounds on the number of variables for the P-hardness of the model checking problem as for the PSPACE-hardness of the validity problem (see above) for the considered intuitionistic logics and their modal companions. Other than for the validity problem, we obtain P-hardness even on the implicational fragments of FPL and BPL with one variable. The PSPACE-hardness of the validity problem on these fragments is open. Since the implicational fragments of IPC and KC for any bounded number of variables have only a finite number of equivalence classes (see [30]), we cannot expect to get P-hardness of model checking on these fragments. We also consider optimality of the P-hardness results in the sense whether model checking with less variables has complexity

¹Unless otherwise stated we expect in the following every Kripke model to be transitive.

below P. We show that model checking for the variable free fragment of FPL drops to LOGCFL, whereas for BPL one can trade the variable in an \vee and keeps P-hardness.

Our results base on a technique we use to show that the model checking problem for the implicational fragment of IPC is P-hard. The variables we use in our construction are essentially needed to measure distances in the model and to mark a certain state. In order to restrict the use of variables, it suffices to express these in a different way. This takes different numbers of variables in the different logics according to their frame properties.

This paper is organized as follows. In Section 2 we introduce the notations for the logics under consideration, and we show P-completeness of a graph accessibility problem for a special case of alternating graphs that will be used for our P-hardness proofs. In Section 3 we consider model checking for the intuitionistic logics KC, IPC, FPL, and BPL. It starts with the P-hardness results (Section 3.1), and closes with the optimality of bounds on the number of variables needed to obtain P-hardness (Section 3.2). In Section 4 the results for the modal companions S4.2, S4, PrL, and K4 follow. The arising completeness results and conclusions are drawn in Section 5. An overview of the results is given in Figures 7 and 8.

2. PRELIMINARIES

Kripke Models. We will consider different propositional logics whose formulas base on a countable set PROP of *propositional variables*. A *Kripke model* is a triple $\mathcal{M} = (U, R, \xi)$, where U is a nonempty and finite set of *states*, R is a binary relation on U , and $\xi : \text{PROP} \rightarrow \mathfrak{P}(U)$ is a function—the *valuation function*. For any variable it assigns the set of states in which this variable is satisfied. (U, R) can also be seen as a directed graph—it is called a *frame* in this context. A frame (U, R) is *reflexive*, if $(x, x) \in R$ for all $x \in U$, it is *irreflexive*, if $(x, x) \notin R$ for all $x \in U$, and it is *transitive*, if for all $a, b, c \in U$, it follows from $(a, b) \in R$ and $(b, c) \in R$ that $(a, c) \in R$. A reflexive and transitive frame is called a *preorder*. If a preorder (U, R) has the additional property that for all $a, b \in U$ there exists a $c \in U$ with $(a, c) \in R$ and $(b, c) \in R$, then (U, R) is called a *directed preorder*.

Modal Propositional Logic. The language \mathcal{ML} of modal logic is the set of all formulas of the form

$$\varphi ::= \perp \mid p \mid \varphi \rightarrow \varphi \mid \Box \varphi,$$

where $p \in \text{PROP}$. As usual, we use the abbreviations $\neg \varphi := \varphi \rightarrow \perp$, $\top := \neg \perp$, $\varphi \vee \psi := (\neg \varphi) \rightarrow \psi$, $\varphi \wedge \psi := \neg(\varphi \rightarrow \neg \psi)$, and $\Diamond \varphi := \neg \Box \neg \varphi$.

The semantics is defined via Kripke models. Given a Kripke model $\mathcal{M} = (U, R, \xi)$ and a state $s \in U$, the *satisfaction relation for modal logics* $\models_{\mathcal{M}}$ is defined as follows.

$$\begin{aligned} \mathcal{M}, s &\not\models_{\mathcal{M}} \perp \\ \mathcal{M}, s &\models_{\mathcal{M}} p && \text{iff } s \in \xi(p), \ p \in \text{PROP}, \\ \mathcal{M}, s &\models_{\mathcal{M}} \varphi \rightarrow \psi && \text{iff } \mathcal{M}, s \not\models_{\mathcal{M}} \varphi \text{ or } \mathcal{M}, s \models_{\mathcal{M}} \psi, \\ \mathcal{M}, s &\models_{\mathcal{M}} \Box \varphi && \text{iff } \forall t \in U \text{ with } (s, t) \in R : \mathcal{M}, t \models_{\mathcal{M}} \varphi. \end{aligned}$$

For $\mathcal{M}, s \models_{\mathcal{M}} \varphi$ we say that formula φ is *satisfied* by model \mathcal{M} in state s .

The modal logic defined in this way is called K and it is the weakest normal modal logic. We will consider the stronger modal logics K4, S4, S4.2, and PrL. The formulas in all these logics are the same as for \mathcal{ML} . Since we are interested in model checking, we use the semantics defined by Kripke models. They will be defined by properties of the frame (U, R) that is part of the model. The semantics of K4 is defined by transitive frames. This means, that a formula α is a theorem of K4 if and only if $\mathcal{M}, w \models_m \alpha$ for all Kripke models \mathcal{M} whose frame is transitive and all states w of \mathcal{M} . The semantics of S4 is defined by preorders, of S4.2 by directed preorders, and of PrL by transitive and irreflexive frames.

Intuitionistic Propositional Logic. The language \mathcal{IL} of intuitionistic logic is essentially the same as that of classical propositional logic, i.e. it is the set of all formulas of the form

$$\varphi ::= \perp \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi,$$

where $p \in \text{PROP}$. As usual, we use the abbreviations $\neg\varphi := \varphi \rightarrow \perp$ and $\top := \neg\perp$. Because of the semantics of intuitionistic logic, one cannot express \wedge or \vee using implication and \perp . Therefore we use \rightarrow instead of \rightarrow .

The semantics is defined via Kripke models $\mathcal{M} = (U, \triangleleft, \xi)$ that fulfil certain restrictions. Firstly, \triangleleft is transitive, and secondly, the valuation function $\xi : \text{PROP} \rightarrow \mathfrak{P}(U)$ is monotone in the sense that for every $p \in \text{PROP}$, $a, b \in U$: if $a \in \xi(p)$ and $a \triangleleft b$, then $b \in \xi(p)$. We will call models that fulfil both these properties *intuitionistic* or *model for BPL*. An intuitionistic model $\mathcal{M} = (U, \triangleleft, \xi)$ where \triangleleft is additionally reflexive (i.e. \triangleleft is a preorder) is called a *model for IPC*. If \triangleleft is a directed preorder, then \mathcal{M} is called a *model for KC*, and if \triangleleft is irreflexive, \mathcal{M} is called a *model for FPL*.

Given an intuitionistic model $\mathcal{M} = (U, \triangleleft, \xi)$ and a state $s \in U$, the *satisfaction relation for intuitionistic logics* \models_i is defined as follows.

$$\begin{aligned} \mathcal{M}, s &\not\models_i \perp \\ \mathcal{M}, s &\models_i p & \text{iff } s \in \xi(p), p \in \text{PROP}, \\ \mathcal{M}, s &\models_i \varphi \wedge \psi & \text{iff } \mathcal{M}, s \models_i \varphi \text{ and } \mathcal{M}, s \models_i \psi, \\ \mathcal{M}, s &\models_i \varphi \vee \psi & \text{iff } \mathcal{M}, s \models_i \varphi \text{ or } \mathcal{M}, s \models_i \psi, \\ \mathcal{M}, s &\models_i \varphi \rightarrow \psi & \text{iff } \forall n \in U \text{ with } s \triangleleft n : \text{ if } \mathcal{M}, n \models_i \varphi \text{ then } \mathcal{M}, n \models_i \psi \end{aligned}$$

An important property of intuitionistic logic is that the monotonicity property of the valuation function also holds for all formulas φ : if $\mathcal{M}, s \models_i \varphi$ then $\forall n$ with $s \triangleleft n$ holds $\mathcal{M}, n \models_i \varphi$.

A formula φ is *satisfied* by an intuitionistic model \mathcal{M} in state s if and only if $\mathcal{M}, s \models_i \varphi$. Basic propositional logic BPL [32] (resp. IPC, KC, FPL [32]) is the set of \mathcal{IL} -formulas that are satisfied by every model for BPL (resp. IPC, KC, FPL) in every state.

Modal Companions. Gödel-Tarski translations map intuitionistic formulas to modal formulas in a way that preserves validity in the different logics. We take the translation 1 from [32], that we call gt and that is defined as follows.

intuitionistic logic	modal companion	frame properties
BPL	K4	transitive
IPC	S4	transitive and reflexive (= preorder)
KC	S4.2	directed preorder
FPL	PrL	transitive and irreflexive

Figure 1: Intuitionistic logics, their modal companions, and the common frame properties.

$$\begin{aligned}
gt(\perp) &:= \perp \\
gt(p) &:= p \wedge \Box p \text{ (for all } p \in \text{PROP)} \\
gt(\alpha \wedge \beta) &:= gt(\alpha) \wedge gt(\beta) \\
gt(\alpha \vee \beta) &:= gt(\alpha) \vee gt(\beta) \\
gt(\alpha \rightarrow \beta) &:= \Box(gt(\alpha) \rightarrow gt(\beta))
\end{aligned}$$

Visser [32] showed that α is valid in FPL if and only if $gt(\alpha)$ is valid in PrL. Therefore, PrL is called *modal companion* of FPL. It is straightforward to see that gt can also be used to show that K4 (resp. S4, S4.2) is a modal companion of BPL (resp. IPC, KC). Figure 1 gives an overview of the intuitionistic logics and their modal companions used here.

Model Checking Problems. This paper examines the model checking problems $L\text{-KMc}$ for logics L whose formulas are evaluated on Kripke models with different properties.

Problem: $L\text{-KMc}$
Input: $\langle \varphi, \mathcal{M}, s \rangle$, where
 φ is a formula for L , $\mathcal{M} = (U, R, \xi)$ is a Kripke model for L , and $s \in U$
Question: Is φ satisfied by \mathcal{M} in state s ?

We assume that formulas and Kripke models are encoded in a straightforward way. This means, a formula is given as a text, and the graph (U, R) of a Kripke model is given by its adjacency matrix that takes $|U|^2$ bits. Therefore, only finite Kripke models can be considered and it can be easily decided whether the model has the order property for the logic under consideration.

Complexity. We assume familiarity with the standard notions of complexity theory as, e.g., defined in [20]. The complexity classes we use in this paper are P (polynomial time) and some of its subclasses. LOGCFL is the class of sets that are logspace many-one reducible to context-free languages. It is also characterized as sets decidable by a nondeterministic Turing machine in polynomial time and logarithmic space with additional use of a stack. L denotes logspace, and NL nondeterministic logspace. To round off the picture, NC^1 (= alternating logarithmic time) is the class for which the model checking problem for classical propositional logic is complete [3], and the model checking problem for IPC_1 is complete for AC^1 (= alternating logspace with logarithmically bounded number of alternations) [19]. The inclusion structure of the classes under consideration is as follows.

$$\text{NC}^1 \subseteq \text{L} \subseteq \text{NL} \subseteq \text{LOGCFL} \subseteq \text{AC}^1 \subseteq \text{P}$$

Fisher and Ladner [10] showed that model checking for modal logic is in P.

Theorem 2.1. [10] K-KMc is in P. □

The notion of reducibility we apply is the logspace many-one reduction \leq_m^{\log} . The Gödel-Tarski translation gt can be seen as such a reduction between the model checking problems for intuitionistic logics and their modal companions, namely $\text{BPL-KMc} \leq_m^{\log} \text{K4-KMc}$, $\text{IPC-KMc} \leq_m^{\log} \text{S4-KMc}$, $\text{KC-KMc} \leq_m^{\log} \text{S4.2-KMc}$, and $\text{FPL-KMc} \leq_m^{\log} \text{PrL-KMc}$. Since gt does not introduce additional variables, the respective reducibilities also hold for the model checking problems for formulas with any restricted number of variables. It therefore follows from Theorem 2.1 that P is an upper bound for all model checking problems for modal respectively intuitionistic logics considered in this paper.

Fragments of Logics. We consider fragments with bounded number of variables or \rightarrow as only connective. The implicational formulas are the formulas with \rightarrow and \perp as only connectives. For an intuitionistic logic L , we use L^\rightarrow to denote the implicational formulas of L , i.e. its *implicational fragment*. L_i denotes its *fragment with i variables*, i.e. the formulas of L with at most i variables. L_i^\rightarrow denotes the implicational fragment with i variables. For modal logics, the (strictly) *implicational fragment* consists of formulas of the form

$$\varphi ::= \perp \mid p \mid \Box(\varphi \rightarrow \varphi).$$

We use the same notation for implicational fragments of modal logics (resp. with bounded numbers of variables) as for intuitionistic logics.

The Gödel-Tarski translation gt does not translate formulas of the implicational fragment of intuitionistic logics into the strictly implicational fragment of modal logics. For the model checking problem, we can use a different translation that preserves satisfaction but does not preserve validity. Let gt' be the translation that is the same as gt but $gt'(p) = p$ for every variable p .

Lemma 2.2. Let α be an \mathcal{IL} -formula, and \mathcal{M} be an intuitionistic model with state s . Then $\mathcal{M}, s \models_i \alpha$ if and only if $\mathcal{M}, s \models_m gt'(\alpha)$. If α is an implicational formula, then $gt'(\alpha)$ is strictly implicational. □

P-complete Problems. Chandra, Kozen, and Stockmeyer [6] have shown that the Alternating Graph Accessibility Problem AGAP is P-complete. In [11] it is mentioned that P-completeness also holds for a bipartite version.

An *alternating graph* $G = (V, E)$ is a bipartite directed graph where $V = V_\exists \cup V_\forall$ are the partitions of V . Nodes in V_\exists are called *existential* nodes, and nodes in V_\forall are called *universal* nodes. The property $apath_G(x, y)$ for nodes $x, y \in V$ expresses that there exists an alternating path through G from node x to node y , and it is defined as follows.

- 1) $apath_G(x, x)$ holds for all $x \in V$
- 2a) for $x \in V_\exists$: $apath_G(x, y)$ if and only if $\exists z \in V_\forall : (x, z) \in E$ and $apath_G(z, y)$
- 2b) for $x \in V_\forall$: $apath_G(x, y)$ if and only if $\forall z \in V_\exists : \text{if } (x, z) \in E \text{ then } apath_G(z, y)$

The problem AGAP consists of directed bipartite graphs G and nodes s, t that satisfy the property $apath_G(s, t)$. Notice that in bipartite graphs existential and universal nodes are strictly alternating.

Problem: AGAP
Input: $\langle G, s, t \rangle$, where G is a directed bipartite graph
Question: does $\text{apath}_G(s, t)$ hold?

Theorem 2.3. [6, 11] AGAP is P-complete. \square

For our purposes, we need an even more restricted variant of AGAP. We require that the graph is *sliced*. An *alternating slice graph* $G = (V, E)$ is a directed bipartite acyclic graph with a bipartitioning $V = V_\exists \cup V_\forall$, and a further partitioning $V = V_1 \cup V_2 \cup \dots \cup V_m$ (m slices, $V_i \cap V_j = \emptyset$ if $i \neq j$) where

$$\begin{aligned} V_\exists &= \bigcup_{i \leq m, i \text{ odd}} V_i, \\ V_\forall &= \bigcup_{i \leq m, i \text{ even}} V_i, \text{ and} \\ E &\subseteq \bigcup_{i=1,2,\dots,m-1} V_i \times V_{i+1}, \text{ i.e. all edges go from slice } V_i \text{ to slice } V_{i+1}. \end{aligned}$$

Finally, we require that all nodes in a slice graph excepted those in the last slice V_m have outdegree > 0 .

Problem: ASAGAP
Input: $\langle G, s, t \rangle$, where $G = (V_\exists \cup V_\forall, E)$ is a slice graph with slices V_1, \dots, V_m ,
and $s \in V_1 \cap V_\exists$, $t \in V_m \cap V_\forall$
Question: does $\text{apath}_G(s, t)$ hold?

It is not hard to see that this version of the alternating graph accessibility problem remains P-complete.

Lemma 2.4. ASAGAP is P-complete.

Proof sketch. ASAGAP is in P, since it is a special case of AGAP. In order to show P-hardness of ASAGAP, it suffices to find a reduction $\text{AGAP} \leq_m^{\log} \text{ASAGAP}$. For an instance $\langle G, s, t \rangle$ of AGAP with graph $G = (V, E)$ where $V = V_\exists \cup V_\forall$ has n nodes, we construct an alternating slice graph $G' = (V', E')$ with $m = 2n$ slices as follows. Let $V'_i = \{\langle v, i \rangle \mid v \in V\}$ for $1 \leq i \leq m$, $V'_\exists = \bigcup_{i \text{ odd}} V'_i$, and $V'_\forall = \bigcup_{i \text{ even}} V'_i$. The edges outgoing from a slice V'_i for odd $i < n$ (existential slice) are

$$E'_i = \left\{ (\langle u, i \rangle, \langle v, i+1 \rangle) \mid (u, v) \in E \text{ and } u \in V_\exists - \{t\} \right\} \cup \left\{ (\langle u, i \rangle, \langle u, i+1 \rangle) \mid u \in V_\forall \cup \{t\} \right\}$$

and for even i (universal slice) accordingly

$$E'_i = \left\{ (\langle u, i \rangle, \langle v, i+1 \rangle) \mid (u, v) \in E \text{ and } u \in V_\forall - \{t\} \right\} \cup \left\{ (\langle u, i \rangle, \langle u, i+1 \rangle) \mid u \in V_\exists \cup \{t\} \right\}.$$

Then $G' = (V'_\exists \cup V'_\forall, E'_1 \cup \dots \cup E'_{m-1})$. The transformation from G to G' can be computed in logarithmic space. It is not hard to see that $\langle G, s, t \rangle \in \text{AGAP}$ if and only if $\langle G', \langle s, 1 \rangle, \langle t, m \rangle \rangle \in \text{ASAGAP}$. \square

Our basic P-hardness proofs of model checking problems will use logspace reductions from ASAGAP. The structural basis can be seen in the proof of the folklore result about K_0 —the fragment of modal logic without variables—that we extend to the strictly implicational fragment K_0^\rightarrow .

Theorem 2.5. *The model checking problem for K_0^\rightarrow is P-hard.*

Proof. First, we give a straightforward transformation from ASAGAP to K_0 -KMC. Second, we turn this into a reduction from $\overline{\text{ASAGAP}}$ to K_0^\rightarrow -KMC.

Let $\langle G, s, t \rangle$ be an instance of ASAGAP, where $G = (V, E)$ is a slice graph with m slices. We construct the model $\mathcal{M}_G := (V, E \cup \{(t, t)\}, \xi)$ and the formula $\varphi_G := \Diamond \Box \Diamond \dots \Box \Diamond (\Diamond \top)$ that consists of a sequence of $m - 1$ alternating modal operators starting with \Diamond that is followed by $\Diamond \top$. Notice that t is the only state in V_m that has a successor, and therefore it is the only state in V_m where $\Diamond \top$ is satisfied. Intuitively speaking, the prefix of $\Diamond \top$ in φ_G that consists of alternating modal operators simulates the alternating path through G from s , and eventually $\Diamond \top$ is satisfied on all the endpoints of this alternating path only if all endpoints equal t . It is not hard to see that an alternating path from s to t exists in G if and only if $\mathcal{M}_G, s \models_m \varphi_G$, i.e. $\langle G, s, t \rangle \in \text{ASAGAP}$ if and only if $\mathcal{M}_G, s \models_m \varphi_G$. Accordingly, $\langle G, s, t \rangle \in \overline{\text{ASAGAP}}$ if and only if $\mathcal{M}_G, s \models_m \neg \varphi_G$, where $\overline{\text{ASAGAP}}$ denotes the complement of ASAGAP.

We now transform $\neg \varphi_G$ into an equivalent formula in the strictly implicational fragment. Using duality of \Diamond and \Box we obtain that $\neg \Diamond \Box \Diamond \dots \Box \Diamond (\Diamond \top)$ is equivalent to $\Box \neg \Box \neg \Box \dots \neg \Box \neg \Box (\Box \perp)$. Every subformula $\Box \neg \alpha$ is equivalent to $\Box (\alpha \rightarrow \perp)$, and the final $\Box (\Box \perp)$ is equivalent to $\Box (\top \rightarrow \Box (\top \rightarrow \perp))$, where $\top \equiv \Box (\perp \rightarrow \perp)$. In this way, $\neg \varphi_G$ can be transformed into the equivalent formula φ'_G that belongs to the strictly implicational fragment. It is straightforward that the mapping $\langle G, s, t \rangle \mapsto \langle \varphi'_G, \mathcal{M}_G, s \rangle$ can be computed in logarithmic space. Since φ'_G contains no variables and belongs to the strictly implicational fragment, this yields $\overline{\text{ASAGAP}} \leq_m^{\log} K_0^\rightarrow$ -KMC, and the P-hardness of K_0^\rightarrow -KMC follows from the P-completeness of ASAGAP (Lemma 2.4) and the closure of P under complement. \square

In general, the slice graph is transformed into a frame (of a Kripke model) to be used in an instance of the model checking problem. Since the semantics of the logics under consideration is defined by Kripke models with frames that are transitive (and reflexive), we need to produce frames that are transitive (and reflexive). The straightforward way would be to take the transitive closure of a slice graph. But this cannot be computed with the resources that are allowed for our reduction functions, i.e. in logarithmic space. Fortunately, slice graphs can easily be made transitive by adding all edges that “jump” from a node to a node that is at least two slices higher. Clearly, the resulting graph is a transitive supergraph of the transitive closure of the slice graph. In order to make the reductions from ASAGAP to the model checking problems work, the valuation function of the Kripke model and the formula that has to be evaluated have to be constructed in a way that “ignores” these edges that jump over a slice.

Definition 2.6. Let $V_{\geq i} = \bigcup_{j=i, i+1, \dots, m} V_j$, and $V_{\leq i} = \bigcup_{j=1, 2, \dots, i} V_j$. The *pseudo-transitive closure* of a slice graph $G = (V, E)$ with m slices $V = V_1 \cup \dots \cup V_m$ is the graph $G' = (V, E')$ where

$$E' := E \cup \bigcup_{i=1, 2, \dots, m-2} (V_i \times V_{\geq i+2}).$$

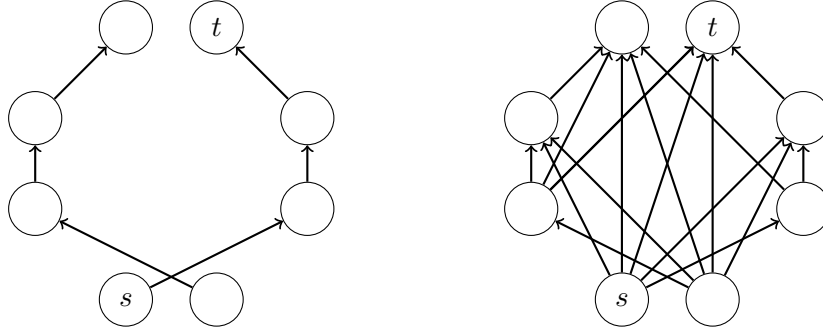


Figure 2: A slice graph and its pseudo-transitive closure

The *reflexive and pseudo-transitive closure* of the slice graph G is the graph $G'' = (V, E'')$ where

$$E'' := E' \cup \{(u, u) \mid u \in V\}.$$

An example for a slice graph and its pseudo-transitive closure is shown in Figure 2.

3. LOWER BOUNDS FOR INTUITIONISTIC LOGICS

We investigate the complexity of the model checking problem for fragments of the intuitionistic logics KC, IPC, FPL, and BPL in Section 3.1. Our basic proof idea is presented in the proof of Theorem 3.1 where we show the P-hardness of KC^{\rightarrow} -KMc. This hardness result carries directly over to IPC^{\rightarrow} -KMc and BPL^{\rightarrow} -KMc. In order to obtain results for fragments with a restricted number of variables we extend the construction from the basic proof. In a first step, we show the P-hardness of model checking for FPL^{\rightarrow} even if we consider formulas with only one variable, i.e. FPL_1^{\rightarrow} -KMc. The same proof works for the P-hardness of BPL_1^{\rightarrow} -KMc. In a second step, we yield P-hardness of BPL_0^{\rightarrow} -KMc. Notice that it remains open whether BPL_0^{\rightarrow} -KMc is P-hard, too. Our last P-hardness result in Section 3.1 shows that KC_2 -KMc and IPC_2 -KMc are P-hard. In Section 3.2 we show that the results for FPL_1^{\rightarrow} -KMc, KC_2 -KMc, and IPC_2 -KMc are optimal in the sense, that with one variable less the model checking problem cannot be P-hard, unless unexpected collapses of complexity classes happen.

3.1. P-hard fragments.

We present the basic construction in the proof of Theorem 3.1, where we show the P-hardness of the model checking problem for the implicational fragment of KC. For this, we use a logspace reduction from ASAGAP to KC^{\rightarrow} -KMc. The P-hardness of the model checking problems for the implicational fragments of IPC and BPL follow straightforwardly.

Theorem 3.1. *The model checking problem for KC^{\rightarrow} is P-hard.*

Proof. We show $ASAGAP \leq_m^{\log} KC^{\rightarrow}$ -KMc. The result then follows from Lemma 2.4.

Let $\langle G, s, t \rangle$ be an instance of ASAGAP. We show how to construct a model \mathcal{M}_G and a formula ψ_G such that $\langle G, s, t \rangle \in ASAGAP$ if and only if $\mathcal{M}_G, s \models \psi_G$. Let the slice

graph $G = (\mathcal{V}, E)$ have m slices, with $\mathcal{V} = V_{\exists} \cup V_{\forall}$, and $V_{\exists} = V_1 \cup V_3 \cup \dots \cup V_{m-1}$, and $V_{\forall} = V_2 \cup V_4 \cup \dots \cup V_m$. We use $V_{\geq i}$ to denote $\bigcup_{j \geq i} V_j$.

In order to use G as a frame of a model for KC, it must be a directed preorder. To get (V, \leq) we build the pseudo-transitive closure of G , add the slice $V_{m+1} := \{top\}$, add edges from every node in \mathcal{V} to top , and build the reflexive closure. It is clear that (V, \leq) can be computed from G in logarithmic space. For simplicity of notation we write $x < y$ or $y > x$ for $x \leq y$ and $x \neq y$, and we also use $x \geq y$ and $x > y$ in the same way. The variables that we will use in our formulas are a_1, \dots, a_{m+1} . Informally, a_i is satisfied in the states of the slices V_{i+1}, \dots, V_{m+1} , further a_m is satisfied in the goal node t , and a_{m+1} is satisfied in top . Define the valuation function ξ by $\xi(a_i) := V_{i+1} \cup \dots \cup V_{m+1}$ (for $i = 1, 2, \dots, m-1$), $\xi(a_m) := \{t, top\}$, and $\xi(a_{m+1}) := \{top\}$. The Kripke model $\mathcal{M}_G = (V, \leq, \xi)$ is a model that satisfies the requirements for KC.

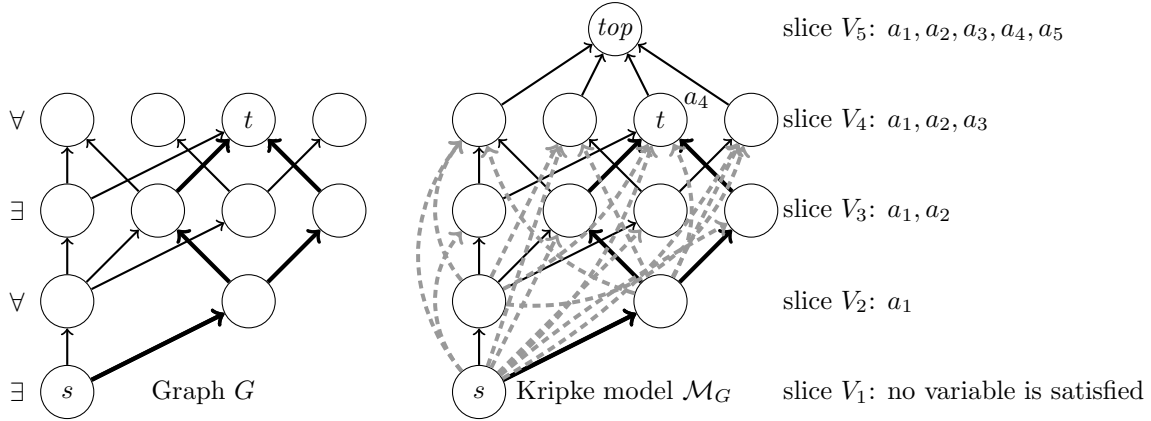


Figure 3: A slice graph G , and the model \mathcal{M}_G as constructed in the proof of Theorem 3.1. The edges to the top node are not drawn, reflexive edges are not drawn, and the pseudo-transitive edges are drawn dashed. The valuation marks the nodes (resp. the slices). The fat edges indicate that $apath_G(s, t)$ holds.

Figure 3 shows a slice graph G with $m = 4$ slices and the Kripke model \mathcal{M}_G that is transformed from it. We will use the formulas ψ_1, \dots, ψ_m in order to express the $apath_G$ property on \mathcal{M}_G .

$$\psi_m := a_m \rightarrow a_{m+1}$$

$$\psi_j := \psi_{j+1} \rightarrow a_j \text{ for } j = m-1, m-2, \dots, 1$$

Next we will show that satisfaction of ψ_i in slice V_i depends only on the edges of the graph G and not on the reflexive and pseudo-transitive edges that were added in order to obtain the Kripke structure.

Claim 1. For all $i = 1, 2, \dots, m-1$ the following holds.

- (1) For all $w \in V_{\geq i+1}$ holds $\mathcal{M}_G, w \models_i \psi_i$.
- (2) For all $w \in V_i$ holds $\mathcal{M}_G, w \models_i \psi_i$ if and only if $\mathcal{M}_G, w \not\models_i \psi_{i+1}$.
- (3) For all $w \in V_i$ holds $\mathcal{M}_G, w \models_i \psi_i$ if and only if $\exists u > w, u \in V_{i+1} : \mathcal{M}_G, u \not\models_i \psi_{i+1}$.

Proof of Claim. For part (1), notice that $\psi_i = (\dots((a_m \rightarrow a_{m+1}) \rightarrow a_{m-1}) \rightarrow \dots \rightarrow a_{i+1}) \rightarrow a_i$. Since $\xi(a_i) = V_{\geq i+1}$, the right-hand side of ψ_i is satisfied in all states in $V_{>i}$. Therefore ψ_i is satisfied in all states in $V_{\geq i}$, too.

Part (2) expresses that ψ_i and ψ_{i+1} behave like the mutual complement in slice V_i , and is shown as follows. Let $w \in V_i$.

$$\begin{aligned}
& \mathcal{M}_G, w \models \psi_i \\
& \Leftrightarrow \forall v \geq w : \text{if } \mathcal{M}_G, v \models \psi_{i+1} \text{ then } \mathcal{M}_G, v \models a_i \quad (\text{semantics of } \rightarrow) \\
& \Leftrightarrow \text{if } \mathcal{M}_G, w \models \psi_{i+1} \text{ then } \mathcal{M}_G, w \models a_i \quad (\text{since } \xi(a_i) = V_{>i}) \\
& \Leftrightarrow \mathcal{M}_G, w \not\models \psi_{i+1} \quad (\text{since } \mathcal{M}_G, w \not\models a_i)
\end{aligned}$$

Part (3) can be proven by proving $\mathcal{M}_G, w \not\models \psi_{i+1}$ if and only if $\exists u > w, u \in V_{i+1} : \mathcal{M}_G, u \not\models \psi_{i+1}$, according to (2). The direction from right to left follows immediately from part (1) and the monotonicity of intuitionistic logic. For the other direction, assume $\forall u > w, u \in V_{i+1} : \mathcal{M}_G, u \models \psi_{i+1}$. Firstly, this yields $\forall u > w : \text{if } \mathcal{M}_G, u \models \psi_{i+2} \text{ then } \mathcal{M}_G, u \models a_{i+1}$ (**), and secondly $\forall u > w, u \in V_{i+1} : \mathcal{M}_G, u \not\models \psi_{i+2}$ (by (2)). From the latter, it follows by the monotonicity property of intuitionistic logic that $\mathcal{M}_G, w \not\models \psi_{i+2}$. Notice that $\mathcal{M}_G, w \not\models a_{i+1}$ by construction of ξ , and therefore we have: if $\mathcal{M}_G, w \models \psi_{i+2}$ then $\mathcal{M}_G, w \models a_{i+1}$. Together with (**) follows $\forall u \geq w : \text{if } \mathcal{M}_G, u \models \psi_{i+2} \text{ then } \mathcal{M}_G, u \models a_{i+1}$. This means $\mathcal{M}_G, w \models \psi_{i+1}$. ■

It is our goal to show that ψ_1 is satisfied in state $s \in V_1$ if and only if graph G has an alternating s - t -path, i.e. $\text{apath}_G(s, t)$. We do this stepwise.

Claim 2. For all $i = 1, 2, \dots, m$ and all $w \in V_i$ holds:

- (1) if i is odd: $\text{apath}_G(w, t)$ if and only if $\mathcal{M}_G, w \models \psi_i$, and
- (2) if i is even: $\text{apath}_G(w, t)$ if and only if $\mathcal{M}_G, w \not\models \psi_i$.

Proof of Claim. We prove the claim by induction on i . The base case $i = m$ considers an even i . Let $w \in V_m$. The following equivalences are straightforward.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow w = t \\
& \Leftrightarrow \mathcal{M}_G, w \not\models a_m \rightarrow a_{m+1} (= \psi_m)
\end{aligned}$$

For the induction step, consider $i < m$. First, assume that i is odd. Then the slice V_i consists of existential nodes. Let $w \in V_i$.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow \exists u, (w, u) \in E : \text{apath}_G(u, t) \quad (\text{definition of } \text{apath}_G) \\
& \Leftrightarrow \exists u > w, u \in V_{i+1} : \mathcal{M}_G, u \not\models \psi_{i+1} \quad (\text{induction hypothesis, construction of } \mathcal{M}_G) \\
& \Leftrightarrow \mathcal{M}_G, w \models \psi_i \quad (\text{Claim 1(3)})
\end{aligned}$$

Second, assume that i is even. Then the slice V_i consists of universal nodes. Let $w \in V_i$.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow \forall u, (w, u) \in E : \text{apath}_G(u, t) \quad (\text{definition of } \text{apath}_G) \\
& \Leftrightarrow \forall u > w, u \in V_{i+1} : \mathcal{M}_G, u \models \psi_{i+1} \quad (\text{induction hypothesis, construction of } \mathcal{M}_G) \\
& \Leftrightarrow \mathcal{M}_G, w \not\models \psi_i \quad (\text{Claim 1(3)}) \quad \blacksquare
\end{aligned}$$

Let $\psi_G := \psi_1$. From Claim 2 it now follows that $\langle G, s, t \rangle \in \text{ASAGAP}$ if and only if $\mathcal{M}_G, s \models \psi_G$, i.e. $\langle \psi_G, \mathcal{M}_G, s \rangle \in \text{KC}^\rightarrow\text{-KMC}$. Since \mathcal{M}_G and ψ_G can be constructed from G using logarithmic space, it follows that $\text{ASAGAP} \leq_m^{\log} \text{KC}^\rightarrow\text{-KMC}$. \square

Clearly, the same lower bound holds for the implicational fragments of IPC and BPL.

Corollary 3.2. *The model checking problem for IPC^\rightarrow and BPL^\rightarrow is P-hard.* \square

The basic construction of the reduction from the above proof can be seen as follows. The frame of the model contains all information about the ASAGAP instance from which it is constructed, but there is some “noise” by the pseudo-transitive (and reflexive) edges. The valuation function gives additional information on the structure of the ASAGAP instance. It says where the goal node t sits, and it allows to check the distances of any state to the upper most slice. The formula puts both parts together. It uses the variables to filter out the original ASAGAP instance and to evaluate it.

If we restrict the number of variables to be used in the formula, we need a different approach to measure the distances of the states to the upper most slice. For irreflexive frames, we can replace the variables by formulas that measure this distance. To distinguish the goal node from the other nodes we use one variable. This yields that $\text{FPL}_1^\rightarrow\text{-KMC}$ is P-hard (Theorem 3.3). In Theorem 3.7 we show that we cannot save this variable. Essentially, in the fragment of FPL without variables we can measure distances, but we cannot do more.

Theorem 3.3. *The model checking problem for FPL_1^\rightarrow is P-hard.*

Proof. We show $\overline{\text{ASAGAP}} \leq_m^{\log} \text{FPL}_1^\rightarrow\text{-KMC}$, where $\overline{\text{ASAGAP}}$ is the complement of ASAGAP. Since P is closed under complement, from Lemma 2.4 follows that $\overline{\text{ASAGAP}}$ is P-complete. Therefore we obtain the P-hardness of $\text{FPL}_1^\rightarrow\text{-KMC}$.

Let $\langle G, s, t \rangle$ with $G = (V, E)$ be an instance of ASAGAP with m slices. From that we construct an $\text{FPL}_1^\rightarrow\text{-KMC}$ instance $\langle \psi, \mathcal{M}, s \rangle$. Let p be the variable that is used in FPL_1^\rightarrow . Let (V, \prec) be the pseudo-transitive closure of G (see Definition 2.6). We define $\mathcal{M} := (V, \prec, \xi)$ with $\xi(p) := \{t\}$. We use p to distinguish t from the other states in slice V_m . Figure 4 shows an example of \mathcal{M} with $m = 4$.

To express the apath_G property we use the formulas $\psi_m, \psi_{m-1}, \dots, \psi_1$ defined as follows.

$$\begin{aligned}
\alpha_m &:= \perp, & \psi_m &:= p \\
\alpha_i &:= \top \rightarrow \alpha_{i+1}, & \psi_i &:= \psi_{i+1} \rightarrow \alpha_{i+1} \quad \text{for } i = m-1, m-2, \dots, 1
\end{aligned}$$

Note that the length of ψ_1 is approximately the sum of the lengths of all α_i with $m \geq i > 1$, hence it is about m^2 . We use the α_i formulas as yardsticks for the slices and the ψ_i formulas for the alternation as we did in the proof of Theorem 3.1. According to Claim 1 we give the following claim. Because of the irreflexivity of \mathcal{M} we do not need the mutual complement property (Claim 1(2)).

Claim 3. For all i with $m \geq i \geq 2$ it holds that

- (1) $\mathcal{M}, w \models_i \alpha_i$ if and only if $w \in V_{\geq i+1}$, and
- (2) for all $w \in V_{i-1}$ it holds that $\mathcal{M}, w \not\models_i \psi_{i-1}$ if and only if $\exists v \in V_i, w \prec v : \mathcal{M}, v \models_i \psi_i$.

Proof of Claim. With induction on i we show (1). For $i = m$ it is trivial because $\alpha_m = \perp$. For the induction step let $w \in W$ and $m > i \geq 2$.

$$\begin{aligned}
& \mathcal{M}, w \models_i \alpha_i \quad (= \top \rightarrow \alpha_{i+1}) \\
& \Leftrightarrow \forall v \in V, w \prec v : \mathcal{M}, v \models_i \alpha_{i+1} \quad (\text{semantics of } \rightarrow) \\
& \Leftrightarrow \forall v \in V, w \prec v : v \in V_{\geq i+2} \quad (\text{induction hypothesis}) \\
& \Leftrightarrow w \in V_{\geq i+1} \quad (\text{construction of } \mathcal{M})
\end{aligned}$$

For (2) consider $w \in V_{i-1}$ with $m \geq i \geq 2$.

$$\begin{aligned}
& \mathcal{M}, w \not\models_i \psi_{i-1} \quad (= \psi_i \rightarrow \alpha_i) \\
& \Leftrightarrow \exists v \in V, w \prec v : \mathcal{M}, v \models_i \psi_i \text{ and } \mathcal{M}, v \not\models_i \alpha_i \quad (\text{semantics of } \rightarrow) \\
& \Leftrightarrow \exists v \in V_i, w \prec v : \mathcal{M}, v \models_i \psi_i \quad (\text{Claim 3(1)}) \quad \blacksquare
\end{aligned}$$

According to Claim 2 we have a similar connection between apath_G and the ψ_i formulas.

Claim 4. For all $i = m, m-1, \dots, 1$ and all $w \in V_i$ it holds that:

- (1) if i is even: $\text{apath}_G(w, t)$ if and only if $\mathcal{M}, w \models_i \psi_i$, and
- (2) if i is odd: $\text{apath}_G(w, t)$ if and only if $\mathcal{M}, w \not\models_i \psi_i$.

Proof of Claim. We prove this claim by induction on i . The base case $i = m$ considers an even i . Let $w \in V_m$. The following equivalences are straightforward.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow w = t \\
& \Leftrightarrow \mathcal{M}, w \models_i p
\end{aligned}$$

The induction step is with the help of Claim 3 similar to the induction step in the proof of Claim 2. (Note that the roles of the even and odd slices are swapped.) We consider $i < m$. First, assume that i is even. Then the slice V_i consists of universal nodes. Let $w \in V_i$.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow \forall v \in V, (w, v) \in E : \text{apath}_G(v, t) \quad (\text{definition of } \text{apath}_G) \\
& \Leftrightarrow \forall v \in V_{i+1}, w \prec v : \mathcal{M}, v \not\models_i \psi_{i+1} \quad (\text{induction hypothesis, construction of } \mathcal{M}) \\
& \Leftrightarrow \mathcal{M}, w \models_i \psi_i \quad (\text{Claim 3(2)})
\end{aligned}$$

Second, assume that i is odd, then the slice V_i consists of existential nodes. Let $w \in V_i$.

$$\begin{aligned}
& \text{apath}_G(w, t) \\
& \Leftrightarrow \exists v \in V, (w, v) \in E : \text{apath}_G(v, t) \quad (\text{definition of } \text{apath}_G) \\
& \Leftrightarrow \exists v \in V_{i+1}, w \prec v : \mathcal{M}, v \models_i \psi_{i+1} \quad (\text{induction hypothesis, construction of } \mathcal{M}) \\
& \Leftrightarrow \mathcal{M}, w \not\models_i \psi_i \quad (\text{Claim 3(2)}) \quad \blacksquare
\end{aligned}$$

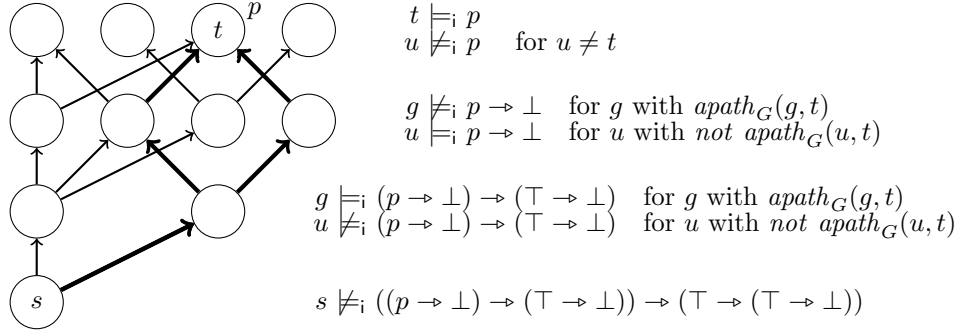


Figure 4: The model \mathcal{M} as constructed from the example instance of ASAGAP in Fig. 3 by the proof of Theorem 3.3. For simplicity, the pseudo-transitive edges are not drawn.

Let $\psi := \psi_1$. It follows from Claim 4 that $\mathcal{M}, s \models_i \psi$ (resp. $\langle \psi, \mathcal{M}, s \rangle \in \text{FPL}_1^\rightarrow\text{-KMc}$) if and only if $\langle G, s, t \rangle \notin \text{ASAGAP}$. Since \mathcal{M} and ψ can be constructed from G using logarithmic space, it follows that $\overline{\text{ASAGAP}} \leq_m^{\log} \text{FPL}_1^\rightarrow\text{-KMc}$. \square

Corollary 3.4. *The model checking problem for BPL_1^\rightarrow is P-hard.* \square

For the fragment of BPL without variables, we can show the P-hardness of model checking only for formulas with the connectives \rightarrow and \vee . Our replacement technique for the last variable costs us the implicationality of the fragment.

Theorem 3.5. *The model checking problem for BPL_0 is P-hard.*

Proof. As in the proof of Theorem 3.3 we show $\overline{\text{ASAGAP}} \leq_m^{\log} \text{BPL}_0\text{-KMc}$. The proof consists of two parts. In the first part we modify the construction that we gave in the proof of Theorem 3.3 in a way that the ψ_i formulas contain two variables but no \perp because we need \perp -free formulas for the second step. In the second step we use a technique from Rybakov [22, Lemma 8] to substitute the variables.

Let $\langle G, s, t \rangle$ with $G = (V, E)$ be an instance from ASAGAP, (V, \prec) be the pseudo-transitive closure of G , and $\mathcal{M} := (V, \prec, \xi)$ with $\xi(p_1) := \{t\}$ and $\xi(p_2) := \emptyset$. Informally, p_2 plays the role of \perp because for all $w \in V$ it holds that $\mathcal{M}, w \not\models_i p_2$. We define the ψ_i formulas as mentioned above.

$$\begin{aligned}
 \theta_m &:= p_2, & \psi_m &:= p_1 \\
 \theta_i &:= \top \rightarrow \theta_{i+1}, & \psi_i &:= \psi_{i+1} \rightarrow \theta_{i+1} \quad \text{for } i = m-1, m-2, \dots, 1
 \end{aligned}$$

For the same reason as in the proof of Theorem 3.3 it holds that

$$\mathcal{M}, s \models_i \psi_1 \Leftrightarrow \langle G, s, t \rangle \notin \text{ASAGAP}.$$

The models $\mathfrak{F}_i = (W_i, R_i)$ for $i = 1, 2, 3$ and the formulas β_1 and β_2 are defined as in the proof of Lemma 8 in [22]. Let for $k = 1, 2, 3$

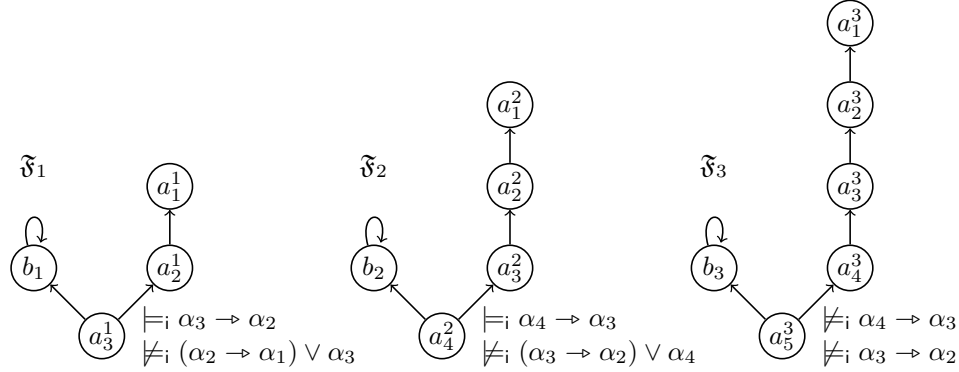


Figure 5: The models \mathfrak{F}_1 , \mathfrak{F}_2 , and \mathfrak{F}_3 . (Transitive edges are not depicted.) It is drawn which parts of β_1 and β_2 are satisfied and which are not. For example let $w \in W^*$ with $(w, a_3^1) \in R^*$, then it holds that $\mathfrak{F}^*, w \not\models_i \beta_1$. Therefore we connect via R^* (resp. R^ξ) every state from $V \setminus \xi(p)$ with a_3^1 and replace p_1 with β_1 in ψ_1 . The models \mathfrak{F}_1 and \mathfrak{F}_2 simulate ξ with technical help of \mathfrak{F}_3 . (For details see the proof of Lemma 8 in [22].)

$$W_k := \{b_k, a_1^k, a_2^k, \dots, a_{k+2}^k\}, \text{ and}$$

$$R_k := \{(b_k, b_k), (a_{k+2}^k, b_k)\} \cup \{(a_i^k, a_j^k) \mid 1 \leq j < i \leq k+2\}.$$

The models are depicted in Figure 5. The formulas β_1 and β_2 are defined as follows. We use the abbreviation $\alpha_1 := \top \rightarrow \perp$ and $\alpha_{i+1} := \top \rightarrow \alpha_i$ for $i \geq 1$.

$$\beta_1 := (\alpha_3 \rightarrow \alpha_2) \rightarrow ((\alpha_2 \rightarrow \alpha_1) \vee \alpha_3)$$

$$\beta_2 := (\alpha_4 \rightarrow \alpha_3) \rightarrow ((\alpha_3 \rightarrow \alpha_2) \vee \alpha_4)$$

We define a $\text{BPL}_0\text{-KMC}$ -instance $\langle \psi_\beta, \mathfrak{F}^*, s \rangle$ with $\mathfrak{F}^* = (W^*, R^*)$.

$$W^* := V \cup W_1 \cup W_2 \cup W_3$$

$$R^\xi := \{(w, a_3^1), (v, a_4^2), (v, a_5^3) \mid w \in W \setminus \{t\}, v \in W\}$$

$$R^* \text{ is the transitive closure of } \prec \cup R_1 \cup R_2 \cup R_3 \cup R^\xi$$

Note that $|W_1 \cup W_2 \cup W_3| = 15$ and \prec is already transitive, hence one can compute the transitive closure in logarithmic space. (We give no valuation function because in BPL_0 models variables are irrelevant.) The connection between β_1 and β_2 and \mathfrak{F}^* is shown and explained in Figure 5. In the following we substitute the variables in ψ_1 .

$$\psi_\beta := \psi_1[p_1/\beta_1][p_2/\beta_2]$$

As Rybakov did in the proof of Lemma 8 in [22] one can show by induction on the construction of ψ that

$$\mathfrak{F}^*, s \models_i \psi_\beta \Leftrightarrow \mathcal{M}, s \models_i \psi_1.$$

(Note that Rybakov shows this only for \perp -free formulas, hence we cannot use the one variable version of ψ_1 from the proof of Theorem 3.3.) It holds that $\langle \psi_\beta, \mathfrak{F}^*, s \rangle \in \text{BPL}_0\text{-KMc}$ if and only if $\langle G, s, t \rangle \notin \text{ASAGAP}$. It follows directly from the construction that this is a logspace reduction. \square

Other than $\text{BPL}_0^{\rightarrow}$ and $\text{FPL}_0^{\rightarrow}$, the implicational fragments of IPC with any bounded number of variables have only a finite number of equivalence classes (see [30]). Therefore they cannot express arbitrary distances in a model. We obtain P-hardness of model checking for the fragment of IPC with two variables, where the formulas consist of arbitrary connectives. The same applies for the fragment of KC with two variables.

The proof uses our basic construction from the proof of Theorem 3.1 and essentially the same replacement of variables as in the proof of [22, Theorem 4] showing that the validity problem for IPC_2 is PSPACE-complete. Whereas there the reduction works in polynomial-time (that suffices to compute transitive closures), our construction must be computable in logarithmic space, and therefore we must deal with the pseudo-transitive closure. Little other technical changes in the proof are needed. For completeness, we present the proof in Appendix A.

Theorem 3.6. *The model checking problem for KC_2 and for IPC_2 is P-hard.* \square

3.2. Optimality of the bounds of the numbers of variables.

The P-hardness of $\text{KC}_2\text{-KMc}$ and $\text{IPC}_2\text{-KMc}$ (Theorem 3.6) is optimal because $\text{KC}_1\text{-KMc} \in \text{NC}^1$ and $\text{IPC}_1\text{-KMc} \in \text{AC}^1$ [19]. In order to show the optimality of the P-hardness of $\text{FPL}_1^{\rightarrow}\text{-KMc}$ (Theorem 3.3), we show that the complexity of $\text{FPL}_0\text{-KMc}$ is below P.

Theorem 3.7. *The model checking problem for FPL_0 is in LOGCFL.*

Proof. Visser [32] gives a systematic construction of representatives of the formula equivalence classes of variable free formulas over irreflexive Kripke models. This enables that every variable free formula can be represented by a small string. We call this string *formula index*. We will show that every state in an FPL_0 model can also be represented by the length of its longest outgoing path. It turns out, that a formula is satisfied in a state if and only if the formula index is greater than the length of the longest path that starts in the state. This yields a LOGCFL algorithm for the model checking problem for FPL_0 .

The formula index of a formula is the index i of the FPL_0 -equivalent² formula α_i from [32, Def. 4.3] defined as follows. Let $i \in \mathbb{N} \cup \{\omega\}$, where $\omega > i$ for all $i \in \mathbb{N}$.

$$\alpha_0 := \perp, \quad \alpha_\omega := \top, \quad \alpha_{i+1} := \top \rightarrow \alpha_i \quad \text{for } i \in \mathbb{N}.$$

Claim 5. [32, Fact 4.4(iii)] Every variable free \mathcal{IL} formula is FPL_0 -equivalent to exactly one α_i .

One can prove the claim with the following case distinction [32, Fact 4.4(ii)].

²Two variable free \mathcal{IL} formulas φ and ψ are FPL_0 -equivalent if for all states w in all FPL_0 models \mathcal{M} it holds that $\mathcal{M}, w \models \varphi \Leftrightarrow \mathcal{M}, w \models \psi$. We denote this as $\varphi \equiv_F \psi$.

$$\begin{aligned}
 &\text{If } \varphi = \perp, && \text{then } \varphi \equiv_F \alpha_0. \\
 &\text{If } \varphi \equiv_F \alpha_a \wedge \alpha_b, && \text{then } \varphi \equiv_F \alpha_{\min\{a,b\}}. \\
 &\text{If } \varphi \equiv_F \alpha_a \vee \alpha_b, && \text{then } \varphi \equiv_F \alpha_{\max\{a,b\}}. \\
 &\text{If } \varphi \equiv_F \alpha_a \rightarrow \alpha_b, && \text{then } \begin{cases} \varphi \equiv_F \alpha_\omega & \text{if } a \leq b \\ \varphi \equiv_F \alpha_{b+1} & \text{if } a > b. \end{cases}
 \end{aligned}$$

If $\varphi \equiv_F \alpha_i$, we call i the *formula index* of φ . In order to analyse the complexity of the formula index computation, we define the following decision problem.

Problem: EQVFORMULA
Input: $\langle \varphi, i \rangle$, where φ is a variable free \mathcal{IL} formula and $i \in \mathbb{N} \cup \{\omega\}$
Question: Is $\alpha_i \equiv_F \varphi$?

Claim 6. EQVFORMULA is in LOGCFL.

Proof of Claim. From the case distinction above one can directly form a recursive algorithm. If $\varphi \equiv_F \alpha_i$ it holds that $i = \omega$ or $i \leq |\varphi|$. ($|\varphi|$ denotes the length of φ .) So every variable value can be stored in logarithmic space. The algorithm walks recursively through the formula and computes the formula index of every subformula once, hence running time is polynomial. All information that are necessary for recursion can be stored on the stack. Therefore the algorithm can be implemented on a polynomial time logspace machine that uses an additional stack i.e. a LOGCFL-machine (even without using nondeterminism). ■

In the following we show that for model checking every FPL_0 model can be reduced to its longest path. Let $\mathcal{M} = (W, R)$ be an FPL_0 model. (Note that we need no valuation function because in FPL_0 models variables are irrelevant.) Therefore we define a function $lp_{\mathcal{M}} : W \rightarrow \mathbb{N}$ that maps a state w to the length of the longest path in \mathcal{M} starting in w .

$$lp_{\mathcal{M}}(w) := \begin{cases} 0, & \text{if } \nexists v \in W : (w, v) \in R \\ \max_{(w,v) \in R} \{lp_{\mathcal{M}}(v)\} + 1, & \text{otherwise} \end{cases}$$

Claim 7.

- (1) Let $\mathcal{M} = (W, R)$ be an FPL_0 model. For every α_i and every state $w \in W$ it holds that $\mathcal{M}, w \models_i \alpha_i$ if and only if $lp_{\mathcal{M}}(w) < i$.
- (2) The following problem is NL-complete: given an FPL_0 model \mathcal{M} , an integer n , and a state w of \mathcal{M} ; does $lp_{\mathcal{M}}(w) = n$ hold?

Proof of Claim. We prove (1) with induction on the formula index i . The cases $i = 0$ and $i = \omega$ are clear. The induction step is shown by the following equivalences.

$$\begin{aligned}
 &\mathcal{M}, w \models_i \alpha_{i+1} \quad (= \top \rightarrow \alpha_i) \\
 &\Leftrightarrow \forall s \in W, (w, s) \in R : \mathcal{M}, s \models_i \alpha_i \quad (\text{semantics of } \rightarrow) \\
 &\Leftrightarrow \forall s \in W, (w, s) \in R : lp_{\mathcal{M}}(s) < i \quad (\text{induction hypothesis}) \\
 &\Leftrightarrow lp_{\mathcal{M}}(w) < i + 1 \quad (\text{irreflexivity of } \mathcal{M})
 \end{aligned}$$

For (2) note that the problem for a given graph G , a node s of G and an integer n to decide whether the longest path in G starting in s has the length n is NL-complete [13]. ■

Algorithm 1 FPL_0 model checking algorithm.

Require: a variable free \mathcal{IL} formula φ , an FPL_0 model \mathcal{M} , and a state w from \mathcal{M}

- 1: guess nondeterministically a formula index $i \in \{0, 1, \dots, |\varphi|\} \cup \{\omega\}$
 - 2: **if** $(\varphi, i) \in \text{EQVFORMULA}$ **then**
 - 3: guess nondeterministically an integer $n < i$
 - 4: **if** $lp_{\mathcal{M}}(w) = n$ **then** accept **else** reject
 - 5: **else** reject
-

Algorithm 1 decides $\text{FPL}_0\text{-KMC}$ with the resources of LOGCFL . In the first two steps we compute the formula index of φ . With Claim 6 it follows that these steps can be done with the resources of LOGCFL . In the next steps the length of the longest path starting in w is guessed and verified. The verification (Step 4) can be done with the resources of NL . The correctness of Step 4 follows from Claim 5 and Claim 7. Altogether Algorithm 1 can be implemented on a nondeterministic polynomial time machine with logarithmic space and an additional stack. These are the resources of LOGCFL . \square

It is not known whether $\text{FPL}_0\text{-KMC}$ is LOGCFL -hard, too. We show NL as lower bound, even for the implicational fragment.

Lemma 3.8. *The model checking problem for FPL_0^\rightarrow is NL -hard.*

Proof sketch. Claim 7 shows that in FPL_0 only the depth of a model can be evaluated by a formula. Accordingly, the α_i formulas can be used to describe the maximal length of a path through a model. This yields a reduction from the longest path problem in acyclic directed graphs to $\text{FPL}_0^\rightarrow\text{-KMC}$. Let $\langle G = (V, E), v \in V, n \in \mathbb{N} \rangle$ be an instance of the longest path problem. Then it holds, that the longest path starting in v has the length n if and only if $G, v \models_i \alpha_{i+1}$ and $G, v \not\models_i \alpha_i$. This follows from Claim 7(1). Since NL is closed under complementation this is a correct reduction. For the NL -completeness of this longest path problem see [13]. \square

4. LOWER BOUNDS FOR MODAL LOGICS

For all P-hard model checking problems for fragments of intuitionistic logics we obtain the same lower bound for their modal companions.

Theorem 4.1. *The model checking problem is P-hard for K4_0 , PrL_1^\rightarrow , S4.2^\rightarrow , K4_1^\rightarrow , and S4^\rightarrow .*

Proof. By Lemma 2.2 this follows from Theorems 3.5, 3.3, and 3.1. \square

From Theorem 3.6 and Lemma 2.2 we obtain that the model checking problem for S4.2_2 —the modal companion of KC_2 —is P-hard. Even though model checking for KC_1 is in NC^1 [19], we can show that one variable suffices to make model checking P-hard for S4.2 .

Theorem 4.2. *The model checking problem for S4.2_1 is P-hard.*

Proof. We show that $\text{ASAGAP} \leq_m^{\log} \text{S4.2}_1\text{-KMC}$. Since ASAGAP is P-hard (Lemma 2.4), the P-hardness of $\text{S4.2}_1\text{-KMC}$ follows.

Let $\langle G, s, t \rangle$ be an instance of ASAGAP , where $G = (V_\exists \cup V_\forall, E)$ is a slice graph with m slices, and $V_\exists = V_1 \cup V_3 \cup \dots \cup V_{m-1}$, and $V_\forall = V_2 \cup V_4 \cup \dots \cup V_m$. We construct a

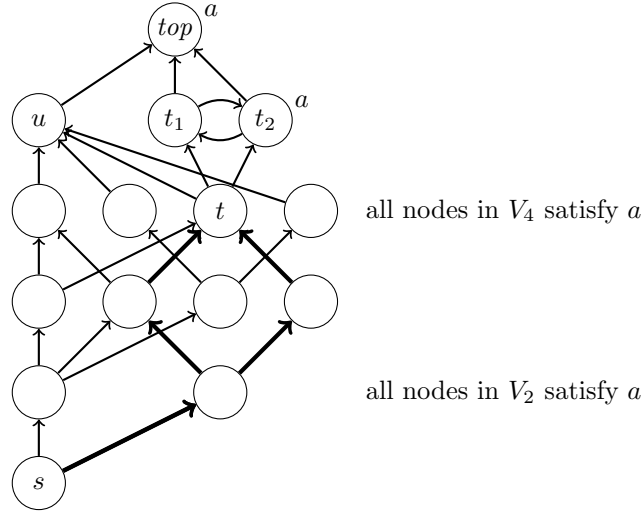


Figure 6: The model \mathcal{M}_G as constructed in the proof of Theorem 4.2 for the ASAGAP instance from Figure 3. Pseudo-transitive edges and reflexive edges are not drawn for simplicity. The valuation marks the nodes (resp. the slices). The fat edges indicate that $apath_G(s, t)$ holds.

Kripke model $\mathcal{M}_G = (U, R, \xi)$ and a formula λ_1 such that $\langle G, s, t \rangle \in \text{ASAGAP}$ if and only if $\langle \lambda_1, \mathcal{M}_G, s \rangle \in \text{S4.2}_1\text{-KMc}$. First, let $G_t = (V, \leq)$ be the pseudo-transitive and reflexive closure of G . Second, we add two slices to G_t , namely $V_{m+1} := \{u, t_1, t_2\}$ and $V_{m+2} := \{top\}$. Third, we add the edges $\{(v, u) \mid v \in V_m\}$ from every node in V_m to u , edges $\{(t, t_1), (t, t_2)\}$ from the goal node $t \in V_m$ to t_1 and to t_2 , and edges $\{(u, top), (t_1, top), (t_2, top)\}$ from every node in V_{m+1} to top . Moreover, in slice V_{m+1} we abstain from the rule that there are no edges between different nodes in the same slice. We also add the edges $\{(t_1, t_2), (t_2, t_1)\}$ between t_1 and t_2 in both directions. Finally, we add pseudo-transitive edges $V_{\leq m-1} \times V_{m+1}$ and $V_{\leq m} \times V_{m+2}$, and reflexive edges to all nodes. Let the graph $G' = (U, R)$ be the graph obtained in this way. Then G' is reflexive, transitive, and every node has an edge to top . Therefore, G' is a directed preorder.

In order to be able to find out in which slice a state is, we mark every even slice $V_2, V_4, \dots, V_m, V_{m+2}$ with the variable a , and in slice V_{m+1} the node t_2 is marked with a . This yields the valuation function ξ to be defined by $\xi(a) := V_2 \cup V_4 \cup \dots \cup V_{m+2} \cup \{t_2\}$, and completes the construction of the Kripke model $\mathcal{M}_G := (U, R, \xi)$. Figure 6 shows an example.

Let $\eta := \neg a \wedge \Diamond(a \wedge \Diamond \neg a)$. We will use that η is satisfied in t_1 , but it is not satisfied in $V_m \cup \{u, t_2, top\}$. The goal node t is the only node in slice V_m that has a successor (namely t_1), in which η is satisfied. We can estimate the slice to which a node belongs using the following formulas δ_i . Let $\delta_m := \Diamond(\neg \eta)$, and for $i = m-1, m-2, \dots, 1$

$$\delta_i := \begin{cases} \Diamond(\neg a \wedge \delta_{i+1}), & \text{if } i \text{ is even,} \\ \Diamond(a \wedge \delta_{i+1}), & \text{if } i \text{ is odd.} \end{cases}$$

Claim 8. Let $x \in V_{\leq m}$ and $i = 1, 2, \dots, m$. Then $\mathcal{M}_G, x \models_m \delta_i$ if and only if $x \in V_{\leq i}$.

Proof of Claim. We proceed by induction on $i = m, m-1, \dots, 1$. The base case $i = m$ is clear, since $\neg\eta$ is satisfied in u and every state in $V_{\leq m}$ has an edge to u . For the induction step consider an arbitrary $i < m$. Let i be odd and $x \in V_{\leq m}$. If $\mathcal{M}_G, x \models_m \delta_i$, then x has a successor y with $\mathcal{M}_G, y \models_m a$ and $\mathcal{M}_G, y \models_m \delta_{i+1}$. By the induction hypothesis we obtain $y \in V_{\leq i+1}$. If $x \neq y$, it follows by the properties of the slice graph that y is a successor of x in a slice “higher” than that of x . The case $x = y$ is not possible because $\mathcal{M}_G, x \models_m \neg a$ and $\mathcal{M}_G, y \models_m a$. Therefore $x \in V_{\leq i}$. For the other proof direction, take any $x \in V_{\leq i}$. The formula δ_i is satisfied in x , if there exists a path of length $m - i + 1$ from x to u in (U, R) , that goes through states that alternatingly satisfy a and $\neg a$. This means, that no edge (v, v) appears on this path. Since every state in $V_{\leq m}$ has a successor in the subsequent slice, such a path exists, and therefore $\mathcal{M}_G, x \models_m \delta_i$. For even i , the proof is similar. ■

The goal state t is the only state in V_m that satisfies $\Diamond\eta$. Using the δ_i formulas to verify an upper bound for the slice of a state, we can now simulate the alternating graph accessibility problem by the following formulas.

Let $\lambda_m := a \wedge \Diamond\eta$ and for $i = m-1, m-2, \dots, 1$

$$\lambda_i := \begin{cases} \neg a \wedge \Diamond(\delta_{i+1} \wedge \lambda_{i+1}), & \text{if } i \text{ is odd,} \\ a \wedge \Box(\delta_{i+1} \rightarrow \lambda_{i+1}), & \text{if } i \text{ is even.} \end{cases}$$

Claim 9. For $i = 1, 2, \dots, m$ and all $x \in V_i$ holds: $\text{apath}_G(x, t)$ if and only if $\mathcal{M}_G, x \models_m \lambda_i$.

Proof of Claim. We prove the claim by induction on i and start with $i = m$. For all $x \in V_m$ holds $\mathcal{M}_G, x \models_m \lambda_m$ if and only if $x = t$, where the latter is the same as $\text{apath}_G(x, t)$. For the induction step, consider an odd $i < m$ first and let $x \in V_i$. We get the following equivalences.

$$\begin{aligned} & \text{apath}_G(x, t) \\ \Leftrightarrow & \exists(x, y) \in E : y \in V_{i+1} \text{ and } \text{apath}_G(y, t) && (\text{definition of } \text{apath}_G) \\ \Leftrightarrow & \exists(x, y) \in R : \mathcal{M}_G, y \models_m \delta_{i+1} \text{ and } \mathcal{M}_G, y \models_m \lambda_{i+1} && (\text{ind. hypoth., Claim 8}) \\ \Leftrightarrow & \mathcal{M}_G, x \models_m \neg a \wedge \Diamond(\delta_{i+1} \wedge \lambda_{i+1}) \quad (= \lambda_i) && (\text{construction of } \mathcal{M}_G) \end{aligned}$$

Second, consider an even $i < m$, and let $x \in V_i$. The following equivalences hold.

$$\begin{aligned} & \text{apath}_G(x, t) \\ \Leftrightarrow & \forall(x, y) \in E : \text{if } y \in V_{i+1} \text{ then } \text{apath}_G(y, t) \\ \Leftrightarrow & \forall(x, y) \in R : \text{if } \mathcal{M}_G, y \models_m \delta_{i+1} \text{ then } \mathcal{M}_G, y \models_m \lambda_{i+1} \\ \Leftrightarrow & \mathcal{M}_G, x \models_m a \wedge \Box(\delta_{i+1} \rightarrow \lambda_{i+1}) \quad (= \lambda_i) \end{aligned}$$

The arguments for the equivalences are the same as above. ■

From Claim 9 it now follows that $\langle G, s, t \rangle \in \text{ASAGAP}$ if and only if $\mathcal{M}_G, s \models_m \lambda_1$, i.e. $\langle \lambda_1, \mathcal{M}_G, s \rangle \in \text{S4.2}_1\text{-KMC}$. Since the construction of \mathcal{M}_G and λ_1 from G can be computed in logarithmic space, it follows that $\text{ASAGAP} \leq_m^{\log} \text{S4.2}_1\text{-KMC}$. □

Note that the reduction in the proof of Theorem 4.2 is not suitable for intuitionistic logics, since the constructed Kripke model lacks the monotonicity property of the variables.

Moreover, in that proof we make extensive use of negation, that would have a very different meaning in intuitionistic logics.

Clearly, the same lower bound holds for the fragment of S4 with one variable.

Corollary 4.3. *The model checking problem for S4₁ is P-hard.* \square

The P-hardness results for S4.2₁-KMc and S4₁-KMc are optimal since the model checking problem for S4₀ is easy to solve. A formula without any variables is either satisfied by every model w.r.t. S4 or it is satisfied by no model. This is because $\Diamond\top$ (resp. $\Box\top$) is satisfied by every state in every model, and $\Diamond\perp$ (resp. $\Box\perp$) is satisfied by no state in every model. Essentially, in order to evaluate a S4₀ formula in some model, the model and the modal operators can be ignored and the remaining classical propositional formula can be evaluated like a classical propositional formula—this problem is NC¹-complete (see [3]).

Lemma 4.4. *The model checking problem for S4₀ and for S4.2₀ are NC¹-complete.* \square

According to Theorem 3.7 we show that the complexity of PrL₀-KMc is below P, namely PrL₀-KMc \in AC¹. Therefore the P-hardness of PrL₁⁺-KMc is optimal in the sense that we cannot save the variable.

Theorem 4.5. *The model checking problem for PrL₀ is in AC¹.*

Proof. We show that every PrL₀ model can be reduced to its longest path. Therefore we define linear models³ $\mathcal{L}_n := (\{0, 1, \dots, n\}, >)$ and use the function $lp_{\mathcal{M}}$, that maps a state to the length of the longest path in its model starting in this state (see the proof of Theorem 3.7). (Note that we give no valuation function because in PrL₀ models variables are irrelevant.) Reinhardt [21] recently showed the upper bound AC¹ for PrL₀ model checking restricted to linear models.

Claim 10. Let $\mathcal{M} = (W, R)$ be a PrL₀ model, $w \in W$, and φ a variable free \mathcal{ML} formula. Then it holds that $\mathcal{M}, w \models_{\mathbf{m}} \varphi$ if and only if $\mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \models_{\mathbf{m}} \varphi$.

Proof of Claim. We show this by induction on the construction φ . The case $\varphi = \perp$ is clear. In the induction step the case $\varphi = \alpha \rightarrow \beta$ is straightforward. Assume that $\varphi = \Box\alpha$.

$$\begin{aligned}
& \mathcal{M}, w \models \varphi \quad (= \Box\alpha) \\
& \Leftrightarrow \forall v \in W, (w, v) \in R : \mathcal{M}, v \models_{\mathbf{m}} \alpha \quad (\text{semantics of } \Box) \\
& \Leftrightarrow \forall v \in W, (w, v) \in R : \mathcal{L}_{lp_{\mathcal{M}}(v)}, lp_{\mathcal{M}}(v) \models_{\mathbf{m}} \alpha \quad (\text{induction hypothesis}) \\
& \Leftrightarrow \forall v \in W, (w, v) \in R : \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(v) \models_{\mathbf{m}} \alpha \quad (\mathcal{L}_{lp_{\mathcal{M}}(v)} \text{ is a submodel of } \mathcal{L}_{lp_{\mathcal{M}}(w)}) \\
& \Leftrightarrow \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \models_{\mathbf{m}} \Box\alpha \quad (= \varphi) \quad (\text{construction of } \mathcal{L}_{lp_{\mathcal{M}}(w)}) \quad \blacksquare
\end{aligned}$$

For a PrL₀ instance $\langle \varphi, \mathcal{M}, w \rangle$ one can compute $lp_{\mathcal{M}}(w)$ with the resources of NL (see [13]). It can be decided whether $\langle \varphi, \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \rangle \in \text{PrL}_0\text{-KMc}$ with the resources of AC¹ [21]. With Claim 10 it holds that $\langle \varphi, \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \rangle \in \text{PrL}_0\text{-KMc}$ if and only if $\langle \varphi, \mathcal{M}, w \rangle \in \text{PrL}_0\text{-KMc}$. Since NL \subseteq AC¹ it holds that PrL₀-KMc \in AC¹. \square

³A frame $\mathcal{M} = (W, R)$ is *linear* if for every $w_1, w_2 \in W$ (with $w_1 \neq w_2$) it holds that either $(w_1, w_2) \in R$ or $(w_2, w_1) \in R$.

It is not known whether AC^1 also is the lower bound of $PrL_0\text{-KMc}$. But from Lemmas 2.2 and 3.8, the lower bound NL follows, even for the strictly implicational fragment.

Lemma 4.6. *The model checking problem for PrL_0^\rightarrow is NL -hard.* \square

Even though we do not know the exact complexity of $FPL_0\text{-KMc}$ and $PrL_0\text{-KMc}$, it is a bit surprising that the $LOGCFL$ upper bound we got for $FPL_0\text{-KMc}$ (Theorem 3.7) is lower than the AC^1 upper bound for $PrL_0\text{-KMc}$ (Theorem 4.5).

5. CONCLUSION

Now we are ready to state the P -completeness results for the model checking problems for intuitionistic logics and their modal companions. Overviews are given in Figures 7 and 8. We start with optimal results for intuitionistic logics.

Theorem 5.1. *The model checking problem is P -complete for FPL_1^\rightarrow , KC_2 , IPC_2 , and BPL_0 . These results are optimal with respect to the number of variables.*

Proof. The upper bound from Theorem 2.1 carries over to all these fragments. The P -hardness for FPL_1^\rightarrow comes from Theorem 3.3, for KC_2 and IPC_2 from Theorem 3.6 and for BPL_0 from Theorem 3.5. The optimality for $FPL_1^\rightarrow\text{-KMc}$ follows from Theorem 3.7 where we show that $FPL_0\text{-KMc}$ is in $LOGCFL$. For $IPC_2\text{-KMc}$ and $KC_2\text{-KMc}$ it follows from [19] where AC^1 -completeness for $IPC_1\text{-KMc}$ and NC^1 -completeness for $KC_1\text{-KMc}$ is shown. \square

For the following results the optimality is still open.

Theorem 5.2. *The model checking problem is P -complete for KC^\rightarrow , IPC^\rightarrow , and BPL_1^\rightarrow .*

Proof. The upper bound from Theorem 2.1 carries over to all these fragments. The P -hardness for KC^\rightarrow comes from Theorem 3.1, for IPC^\rightarrow from Corollary 3.2, and for BPL_1^\rightarrow from Corollary 3.4. \square

It is known that the validity problem for IPC^\rightarrow even without using \perp [25, 4, 28], for FPL^\rightarrow and BPL^\rightarrow [4], and for IPC_2 , FPL_1 , and BPL_0 [22] is $PSPACE$ -complete. We show for all these fragments that model checking is P -complete. Even more, for the implicational fragments FPL_1^\rightarrow and BPL_1^\rightarrow with only one variable we reach P -completeness of model checking. Notice that no $PSPACE$ -hardness results for the validity problem for implicational fragments with a bounded number of variables are known.

Our P -completeness results for $KC^\rightarrow\text{-KMc}$ and $IPC^\rightarrow\text{-KMc}$ hold also for the purely implicational fragments, i.e. KC^\rightarrow and IPC^\rightarrow without using \perp (resp. negation). But what happens if one bounds the number of variables in the implicational fragments? The model checking problem for IPC_1^\rightarrow is NC^1 -complete [19] but for IPC_i^\rightarrow with $i > 1$ it is open whether the complexity is below P . The fragments IPC_i^\rightarrow have finitely many equivalence classes of formulas and models [30, 8]. This equivalence class can be obtained with the resources of NC^1 , using a straightforward extension of the Boolean formula evaluation algorithm of Buss [3]. This might indicate an upper bound lower than P for the model checking problem. But it is not clear how hard it is to obtain the equivalence class of a given model.

Another interesting open question is the complexity of $BPL_0^\rightarrow\text{-KMc}$. We expect the P -completeness of $BPL_1^\rightarrow\text{-KMc}$ to be optimal. But in contrast to IPC_i^\rightarrow even BPL_0^\rightarrow has infinitely many equivalence classes of formulas, because FPL_0^\rightarrow already has it [32]. For FPL_0 , every equivalence class is represented by an implicational formula (see proof of Theorem 3.7).

	number of variables			
	unbounded	2	1	0
BPL	P-complete [→]			P-complete
FPL	P-complete [→]			in LOGCFL NL-hard [→]
IPC	P-complete [→]	P-complete	AC ¹ -complete[19]	NC ¹ -complete[19]
KC	P-complete [→]	P-complete	NC ¹ -complete[19]	

Figure 7: Complexity of the model checking problem for intuitionistic logics.
(The [→] indicates that the result holds for the implicational fragment.)

For $\text{BPL}_0^{\rightarrow}$, it is clear that there are more equivalence classes, but it is open whether they can easily be represented.

For the modal companions we conclude the following and start with the optimal results.

Theorem 5.3. *The model checking problem is P-complete for $\text{PrL}_1^{\rightarrow}$, S4.2_1 , S4_1 , and K4_0 . These results are optimal with respect to the number of variables.*

Proof. For all these fragments the upper bound comes from Theorem 2.1. The P-hardness for $\text{PrL}_1^{\rightarrow}$, and K4_0 comes from Theorem 4.1, for S4.2_1 from Theorem 4.2, and for S4_1 from Corollary 4.3. The optimality for $\text{PrL}_1^{\rightarrow}$ -KMc follows from Theorem 4.5 where we show $\text{PrL}_0\text{-KMc} \in \text{AC}^1$. For $\text{S4.2}_1\text{-KMc}$ and $\text{S4}_1\text{-KMc}$ it follows from Lemma 4.4 where NC^1 -completeness for $\text{S4.2}_0\text{-KMc}$ and $\text{S4}_0\text{-KMc}$ is shown. \square

Notice that $\text{IPC}_1\text{-KMc}$ and $\text{KC}_1\text{-KMc}$ are the only cases where model checking for intuitionistic logics is easier than for its modal companions $\text{S4}_1\text{-KMc}$ and $\text{S4.2}_1\text{-KMc}$.

For the following results the optimality is still open.

Theorem 5.4. *The model checking problem is P-complete for $\text{S4.2}^{\rightarrow}$, S4^{\rightarrow} , and $\text{K4}_1^{\rightarrow}$.*

Proof. For all these fragments the upper bound comes from Theorem 2.1 and the P-hardness comes from Theorem 4.1. \square

Completeness results for $\text{S4.2}^{\rightarrow}\text{-KMc}$ and $\text{S4}^{\rightarrow}\text{-KMc}$ with a bounded number of variables and for $\text{K4}_0^{\rightarrow}$ are still open.

Another semantics for intuitionistic logics is the class of finite trees that are reflexive and transitive. This is a subclass of the intuitionistic Kripke models we used and also sound and complete for IPC. It is open whether the model checking problem for IPC over this tree-semantics is P-hard or below P, and it also remains open for the other P-complete model checking problems of this work.

Acknowledgements. The authors thank Vitezslav Svejdar for helpful discussions about intuitionistic logic, and Thomas Schneider for his support. The authors specially thank an anonymous referee of the Workshop on Reachability Problems 2010 for her/his idea that led to Theorem 4.2. The authors like to thank also the anonymous referees of LMCS for their helpful comments.

	number of variables		
	unbounded	1	0
K	P-complete \rightarrow		
K4	P-complete \rightarrow	P-complete	
PrL	P-complete \rightarrow	in AC ¹ NL-hard \rightarrow	
S4	P-complete \rightarrow	P-complete	NC ¹ -complete
S4.2	P-complete \rightarrow	P-complete	NC ¹ -complete

Figure 8: Complexity of the model checking problem for the modal companions.
(The \rightarrow indicates that the result holds for the strictly implicational fragment.)

REFERENCES

- [1] G. S. Boolos. *The Logic of Provability*. Cambridge University Press, 1993.
- [2] F. Bou. Complexity of strict implication. In *Advances in Modal Logic* 5, pages 1–16, 2005.
- [3] S. R. Buss. The Boolean formula value problem is in ALOGTIME. In *Proc. 19th STOC*, pages 123–131. ACM Press, 1987.
- [4] A. V. Chagrov. On the complexity of propositional logics. In *Complexity Problems in Mathematical Logic*, pages 80–90. Kalinin State University, 1985. In Russian.
- [5] A. V. Chagrov and M. N. Rybakov. How many variables does one need to prove PSPACE-hardness of modal logics. In *Advances in Modal Logic*, volume 4, pages 71–82, 2002.
- [6] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28:114–133, 1981.
- [7] D. H. de Jongh. The maximality of the intuitionistic predicate calculus with respect to Heyting’s arithmetic. *The Journal of Symbolic Logic*, 36:606, 1970.
- [8] G. R. R. de Lavalette, A. Hendriks, and D. H. de Jongh. Intuitionistic implication without disjunction. *Journal of Logic and Computation*. To appear, available at <http://dx.doi.org/10.1093/logcom/exq058>.
- [9] M. Dummett and E. Lemmon. Modal logics between S4 and S5. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 14(24):250–264, 1959.
- [10] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and Systems Sciences*, 18(2):194–211, 1979.
- [11] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, New York, 1995.
- [12] J. Hudelmaier. An $O(n \log n)$ -space decision procedure for intuitionistic propositional logic. *Journal of Logic and Computation*, 3(1):63–75, 1993.
- [13] A. Jakoby and T. Tantau. Logspace algorithms for computing shortest and longest paths in series-parallel graphs. In *FSTTCS*, volume 4855 of *Lecture Notes in Computer Science*, pages 216–227, 2007.
- [14] S. A. Kripke. Semantical analysis of modal logic I. Normal propositional calculi. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
- [15] S. A. Kripke. Semantical analysis of intuitionistic logic I. In *Proc. of the 8th Logics Colloquium*, pages 92–130, 1965.
- [16] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6(3):467–480, 1977.
- [17] J. C. C. McKinsey and A. Tarski. The algebra of topology. *Annals of Mathematics*, 45:141–191, 1944.

- [18] M. Mundhenk and F. Weiß. The complexity of model checking for intuitionistic logics and their modal companions. In *Proc. 4th Int. Workshop on Reachability Problems*, volume 6227 of *LNCs*, pages 146–160. Springer, 2010.
- [19] M. Mundhenk and F. Weiß. The model checking problem for intuitionistic propositional logic with one variable is AC^1 -complete. In *Proc. 28th STACS*, volume 9 of *LIPICs*, pages 368–379. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.
- [20] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.
- [21] K. Reinhardt. Model checking for PrL_0 on linear frames is in AC^1 , 2011. Personal communication.
- [22] M. N. Rybakov. Complexity of intuitionistic and Visser’s basic and formal logics in finitely many variables. In *Advances in Modal Logic 6*, pages 393–411. College Publications, 2006.
- [23] A. Sistla and E. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.
- [24] E. Spaan. *Complexity of Modal Logics*. PhD thesis, Department of Mathematics and Computer Science, University of Amsterdam, 1993.
- [25] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theor. Comput. Sci.*, 9:67–72, 1979.
- [26] V. Svejdar. On provability logic. *Nordic Journal of Philosophical Logic*, 4(2):95–116, 2000.
- [27] V. Svejdar. The decision problem of provability logic with only one atom. *Arch. Math. Log.*, 42(8):763–768, 2003.
- [28] V. Svejdar. On the polynomial-space completeness of intuitionistic propositional logic. *Arch. Math. Log.*, 42(7):711–716, 2003.
- [29] A. Tarski. Der Aussagenkalkül und die Topologie. *Fundamenta Mathematicae*, 31:103–134, 1938.
- [30] A. Urquhart. Implicational formulas in intuitionistic logic. *Journal of Symbolic Logic*, 39(4):661–664, 1974.
- [31] D. van Dalen. *Logic and Structure*. Springer, Berlin, Heidelberg, 4th edition, 2004.
- [32] A. Visser. A propositional logic with explicit fixed points. *Studia Logica*, 40:155–175, 1980.

APPENDIX A.

Theorem 3.6. *The model checking problem for KC_2 and for IPC_2 is P-hard.*

Proof. We show $IPC^{\rightarrow}\text{-}KMc \leq_m^{\log} KC_2\text{-}KMc$. Then P-hardness for $KC_2\text{-}KMc$ and $IPC_2\text{-}KMc$ follows from Corollary 3.2. The construction is similar to the one given by Rybakov [22, Theorem 4] for the PSPACE-completeness of the validity problem for IPC_2 . First of all we construct formulas with two variables which can be used for replacing the variables in arbitrary \mathcal{IL} formulas. We call them *replacement formulas*. Then we give *generic models* which have for every replacement formula a unique maximal refuting state⁴. For a given instance of $IPC\text{-}KMc^{\rightarrow}$ we transform the formula by replacing the variables with the replacement formulas and as model we take the union of the given model and a suitable generic model. This union eventually is a KC_2 model.

The construction—especially the base of the inductive definition of the replacement formulas—is very technical. Let p and q be the variables used in KC_2 . Figure 9 shows the top of the generic model. There, one can see in which states the variables p and q are satisfied (due to the valuation function of the model), and which are the maximal refuting states of the formulas to be defined in the sequel. The essential idea is that every replacement formula has exactly one state that is its maximal refuting state.

Construction of the replacement formulas. The following formulas are the base for the inductive definition of the replacement formulas.

⁴ In $\mathcal{M} = (W, R, \xi)$ the state w refutes φ if $\mathcal{M}, w \not\models_i \varphi$. A state $w \in W$ is a *maximal refuting state* of φ if for all $v \in W \setminus \{w\}$ with $(w, v) \in R$ it holds that $\mathcal{M}, v \models_i \varphi$.

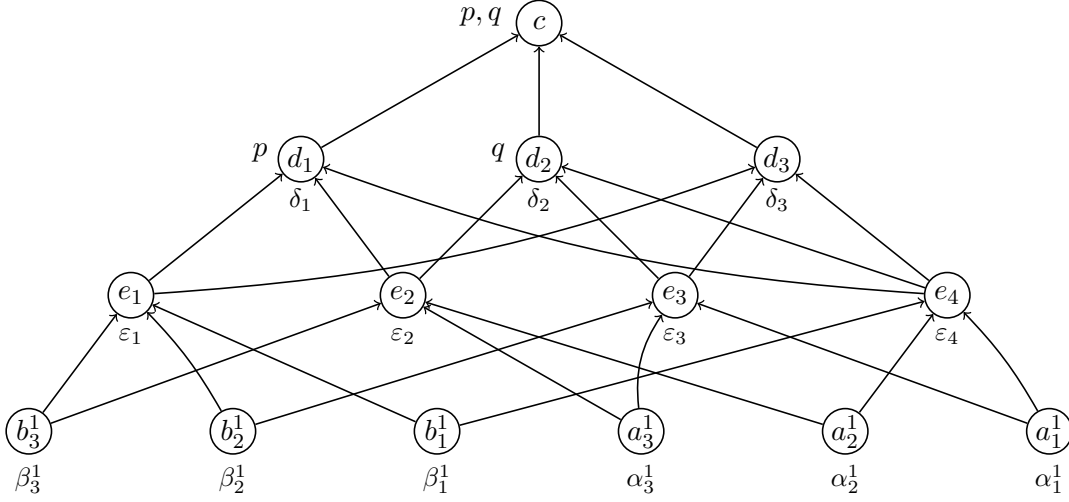


Figure 9: This is the top of the generic model with states from $W_0 \cup W_1$. Every state is labelled below with the formula that it maximally refutes. (Transitive and reflexive edges are not depicted.)

$$\begin{aligned}
 \delta_1 &:= p \rightarrow q & \delta_2 &:= q \rightarrow p & \delta_3 &:= p \vee q \\
 \varepsilon_1 &:= \delta_2 \rightarrow (\delta_1 \vee \delta_3) & \varepsilon_3 &:= \delta_1 \rightarrow (\delta_2 \vee \delta_3) \\
 \varepsilon_2 &:= \delta_3 \rightarrow (\delta_1 \vee \delta_2) & \varepsilon_4 &:= (\varepsilon_1 \wedge \varepsilon_2 \wedge \varepsilon_3) \rightarrow (\delta_1 \vee \delta_2 \vee \delta_3)
 \end{aligned}$$

Using these formulas, the first replacement formulas can be defined as follows.

$$\begin{aligned}
 \alpha_1^1 &:= (\varepsilon_1 \wedge \varepsilon_2) \rightarrow (\varepsilon_3 \vee \varepsilon_4) & \beta_1^1 &:= (\varepsilon_2 \wedge \varepsilon_3) \rightarrow (\varepsilon_1 \vee \varepsilon_4) \\
 \alpha_2^1 &:= (\varepsilon_1 \wedge \varepsilon_3) \rightarrow (\varepsilon_2 \vee \varepsilon_4) & \beta_2^1 &:= (\varepsilon_2 \wedge \varepsilon_4) \rightarrow (\varepsilon_1 \vee \varepsilon_3) \\
 \alpha_3^1 &:= (\varepsilon_1 \wedge \varepsilon_4) \rightarrow (\varepsilon_2 \vee \varepsilon_3) & \beta_3^1 &:= (\varepsilon_3 \wedge \varepsilon_4) \rightarrow (\varepsilon_1 \vee \varepsilon_2)
 \end{aligned}$$

We call the upper index the *level*. The formulas on the next levels will be defined inductively. First we define $n_1 := 3$ and $n_{k+1} := |P_k|$ where $P_k := \{(x, y) \mid 2 \leq x, y \leq n_k\}$. With induction on k one can show that $|P_k| = (n_k - 1)^2$. On level k we define α_i^k and β_i^k for $i = 1, 2, \dots, n_k$. For the step from level k to level $k+1$ we need an encoding $\langle \cdot, \cdot \rangle_k$ from P_k to $\{1, 2, \dots, (n_k - 1)^2\}$ that is easy to compute and easy to decode. For example one can use the following: $\langle \cdot, \cdot \rangle_k$ maps (i, j) to $(j - 1) + (n_k - 1) \cdot (i - 2)$ for $2 \leq i, j \leq n_k$. For $k \geq 1$ the inductive definition is as follows. Let $i, j \in \{2, 3, \dots, n_k\}$.

$$\begin{aligned}
 \alpha_{\langle i, j \rangle_k}^{k+1} &:= \alpha_1^k \rightarrow (\beta_1^k \vee \alpha_i^k \vee \beta_j^k) \\
 \beta_{\langle i, j \rangle_k}^{k+1} &:= \beta_1^k \rightarrow (\alpha_1^k \vee \alpha_i^k \vee \beta_j^k)
 \end{aligned}$$

Construction of the generic models. For $t \geq 1$ we define the generic models $\mathcal{M}_t^S = (W_t^S, R_t^S, \xi^S)$.

$$\begin{aligned}
W_0 &:= \{c, d_1, d_2, d_3, e_1, e_2, e_3, e_4\} \\
W_k &:= \{a_i^k, b_i^k \mid 1 \leq i \leq n_k\} \text{ for } 1 \leq k \leq t \\
W_t^S &:= \bigcup_{l=0}^t W_l
\end{aligned}$$

In the following we give R_t^S . The accessibility relation R_{Top} of the first layers is shown in Figure 9. (Certainly we use the transitive and reflexive closure of the depicted edges.) For states from level ≥ 2 the accessibility relation will be defined as follows. Let $1 \leq k \leq t-1$.

$$\begin{aligned}
R_{k+1}^a &:= \{(a_{\langle i,j \rangle_k}^{k+1}, b_1^k), (a_{\langle i,j \rangle_k}^{k+1}, a_i^k), (a_{\langle i,j \rangle_k}^{k+1}, b_j^k) \mid 2 \leq i, j \leq n_k\} \\
R_{k+1}^b &:= \{(b_{\langle i,j \rangle_k}^{k+1}, a_1^k), (b_{\langle i,j \rangle_k}^{k+1}, a_i^k), (b_{\langle i,j \rangle_k}^{k+1}, b_j^k) \mid 2 \leq i, j \leq n_k\} \\
R' &:= R_{Top} \cup \bigcup_{l=2}^t (R_l^a \cup R_l^b).
\end{aligned}$$

In order to make the accessibility relation transitive, we add pseudo-transitive edges. Every state in a level is connected to every state at least two levels below.

$$T_k := W_k \times \left(\bigcup_{l=0}^{k-2} W_l \right) \text{ for } k \geq 2$$

T is the union of all pseudo-transitive edges.

$$T := \bigcup_{l=2}^t T_l$$

We define the accessibility relation R_t^S as follows.

R_t^S is the reflexive closure of $T \cup R'$.

Figure 10 shows a cutout of \mathcal{M}_t^S . The valuation function ξ^S is defined as follows (see Figure 9).

$$\begin{aligned}
\xi^S(p) &:= \{c, d_1\} \\
\xi^S(q) &:= \{c, d_2\}
\end{aligned}$$

The goal of the construction is that α_i^k (resp. β_i^k) is not satisfied exactly in the states that see a_i^k (resp. b_i^k).

Claim 11. Let w be a state of \mathcal{M}_t^S . Then for all $1 \leq k \leq t$ and $i \leq n_k$ it holds that $\mathcal{M}_t^S, w \not\models \alpha_i^k \Leftrightarrow (w, a_i^k) \in R_t^S$ and $\mathcal{M}_t^S, w \not\models \beta_i^k \Leftrightarrow (w, b_i^k) \in R_t^S$.

The proof can be proceeded by an induction on k (similar as [22, Lemma 5]). Hence for every formula α_i^k and β_i^k exists a unique maximal state in \mathcal{M}_t^S that refutes this formula.

Reduction from IPC^{\rightarrow} model checking problem. For a given instance $\langle \varphi, \mathcal{M}, w \rangle$ of IPC^{\rightarrow} -KMc we show how to translate \mathcal{M} and φ into \mathcal{M}^2 —a model over two variables—and φ^2 —a formula with two variables. Let φ be a formula with variables v_1, v_2, \dots, v_m and $\mathcal{M} = (W, R, \xi)$ a model. We choose the smallest $k > 1$ such that $n_k > m$. To define φ^2 we replace every occurrence of v_i in φ by $\alpha_i^k \vee \beta_i^k$.

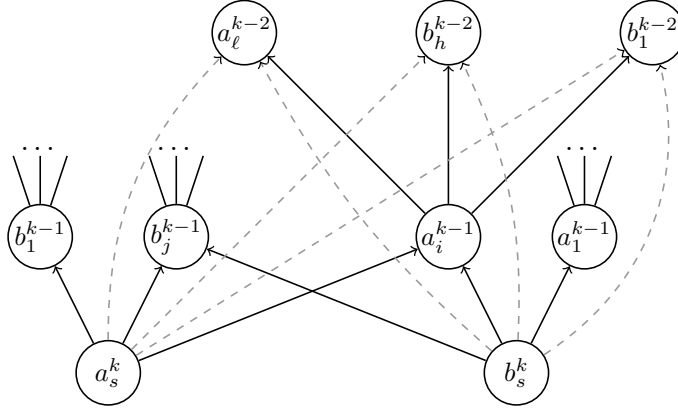


Figure 10: This is a cutout of the levels $k-2$, $k-1$ and k of a generic model \mathcal{M}_t^S where $s = \langle i, j \rangle_{k-1}$, $i = \langle \ell, h \rangle_{k-2}$, and $k \leq t$. The dashed grey edges are the pseudo-transitive edges. As we show in Claim 11 for example a_i^{k-1} is the maximal refuting state for α_i^{k-1} . (Reflexive edges are not depicted.)

$$\varphi^2 := \varphi[v_1/\alpha_1^k \vee \beta_1^k][v_2/\alpha_2^k \vee \beta_2^k] \dots [v_m/\alpha_m^k \vee \beta_m^k]$$

Since $k \leq 1 + \log(m)$ one can construct φ^2 in logspace. We build the translation $\mathcal{M}^2 = (W^2, R^2, \xi^2)$ as a union of \mathcal{M} and \mathcal{M}_k^S .

$$W^2 := W \cup W_k^S$$

The accessibility relation R^2 is constructed such that if $w \notin \xi(v_i)$, then $(w, a_i^k) \in R^2$ and $(w, b_i^k) \in R^2$. Hence w refutes $\alpha_i^k \vee \beta_i^k$ —the translation of v_i .

$$R_\xi := \{ (w, a_i^k), (w, b_i^k) \mid w \in W \setminus \xi(v_i) \} \cup \{ (w, a_{m+1}^k), (w, b_{m+1}^k) \mid w \in W \}$$

In order to make R^2 transitive and give a logspace computable construction we connect every state of \mathcal{M} with every state in \mathcal{M}^S on level $k-1$ and below.

$$R_{trans} := W \times \bigcup_{l=0}^{k-1} W_l$$

We define the accessibility relation R^2 as follows.

$$R^2 \text{ is the reflexive closure of } R_k^S \cup R \cup R_\xi \cup R_{trans}.$$

As valuation function we use

$$\xi^2 := \xi^S.$$

The valuation function ξ of \mathcal{M} is simulated by the edges between \mathcal{M} and \mathcal{M}^S from R_ξ . The model \mathcal{M}^2 is a KC_2 model because for every state $u \in W^2$ it holds that $(u, c) \in R^2$. (The proof of the following claim bases on the proof of [22, Lemma 7].)

Claim 12. For all $w \in W$ it holds that $\mathcal{M}, w \models \varphi$ if and only if $\mathcal{M}^2, w \models \varphi^2$.

Proof of Claim. We prove this by induction on the construction of φ . For the initial step let $\varphi = v_l$ be a variable with $1 \leq l \leq m$, hence $\varphi^2 = \alpha_l^k \vee \beta_l^k$. If $\mathcal{M}, w \not\models v_l$, then w is via R^2 (resp. R_ξ) connected to a_l^k and b_l^k . With Claim 11 it follows that $\mathcal{M}^2, w \not\models \alpha_l^k \vee \beta_l^k$. Now assume that $\mathcal{M}^2, w \not\models \alpha_l^k \vee \beta_l^k$ with $w \in W$. Since

$$\begin{aligned}\alpha_l^k &= \alpha_1^{k-1} \rightarrow (\beta_1^{k-1} \vee \alpha_i^{k-1} \vee \beta_j^{k-1}) \quad \text{and} \\ \beta_l^k &= \beta_1^{k-1} \rightarrow (\alpha_1^{k-1} \vee \alpha_i^{k-1} \vee \beta_j^{k-1})\end{aligned}$$

for $\langle i, j \rangle_{k-1} = l$ it holds that there are some states $w', w'' \in W^2$ with $(w, w') \in R^2$ and $(w, w'') \in R^2$ and

$$\begin{aligned}\mathcal{M}^2, w' &\models \alpha_1^{k-1} \quad \text{and} \quad \mathcal{M}^2, w' \not\models \beta_1^{k-1} \vee \alpha_i^{k-1} \vee \beta_j^{k-1} \quad \text{and} \\ \mathcal{M}^2, w'' &\models \beta_1^{k-1} \quad \text{and} \quad \mathcal{M}^2, w'' \not\models \alpha_1^{k-1} \vee \alpha_i^{k-1} \vee \beta_j^{k-1}.\end{aligned}$$

From Claim 11 follows that $\mathcal{M}^2, a_{m+1}^k \not\models \beta_1^{k-1}$ and $\mathcal{M}^2, b_{m+1}^k \not\models \alpha_1^{k-1}$. Hence it follows for every $u \in W$ that $\mathcal{M}^2, u \not\models \alpha_1^{k-1}$ and $\mathcal{M}^2, u \not\models \beta_1^{k-1}$ because $(u, a_{m+1}^k) \in R^2$ and $(u, b_{m+1}^k) \in R^2$. Therefore $w', w'' \in W_k^S$. Furthermore note that w' and w'' are in level k of \mathcal{M}_k^S because w' refutes α_l^k and w'' refutes β_l^k and with Claim 11 it follows that $w' = a_l^k$ and $w'' = b_l^k$. From $(w, a_l^k) \in R^2$, $(w, b_l^k) \in R^2$, and the construction of R^2 it follows that $w \notin \xi(v_l)$. Hence $\mathcal{M}, w \not\models v_l$.

For the induction step let $\varphi = \gamma \star \delta$ with $\star \in \{\wedge, \vee, \rightarrow\}$. We show that $\mathcal{M}^2, w \models (\gamma \star \delta)^2$ if and only if $\mathcal{M}, w \models \gamma \star \delta$. (Note that $(\gamma \star \delta)^2 = \gamma^2 \star \delta^2$.) For the cases that $\star = \wedge$ and $\star = \vee$ this follows directly from the definition of the satisfaction relation \models . Now consider $\varphi = \gamma \rightarrow \delta$ and $\mathcal{M}, w \not\models \varphi$. Then there is some state $w' \in W$ with $\mathcal{M}, w' \models \gamma$ and $\mathcal{M}, w' \not\models \delta$. By induction hypothesis it follows that $\mathcal{M}^2, w' \models \gamma^2$ and $\mathcal{M}^2, w' \not\models \delta^2$. Hence $\mathcal{M}^2, w \not\models \varphi^2$. For the other proof direction let $w \in W$ with $\mathcal{M}^2, w \not\models \varphi^2$. Then there is a $w' \in W^2$ with $(w, w') \in R^2$ and $\mathcal{M}^2, w' \models \gamma^2$ and $\mathcal{M}^2, w' \not\models \delta^2$. The formulas $\alpha_1^k \vee \beta_1^k, \alpha_2^k \vee \beta_2^k, \dots, \alpha_m^k \vee \beta_m^k$ are satisfied in every state of level k and below, because every state in level k refutes exactly one α_i^k respectively β_i^k formula. (The states below level k satisfy all formulas on level k .) In δ^2 every variable v_i from δ is replaced by the disjunction of an $\alpha_i^k \vee \beta_i^k$. Hence δ^2 is satisfied in every state in W_k^S . In order that w' refutes δ^2 , it holds that $w' \in W$. By induction hypothesis we obtain that $\mathcal{M}, w' \models \gamma$ and $\mathcal{M}, w' \not\models \delta$. From $w, w' \in W$ and $(w, w') \in R^2$ it follows that $(w, w') \in R$. Hence $\mathcal{M}, w \not\models \varphi$. ■

The reduction function is the mapping

$$\langle \varphi, \mathcal{M}, w \rangle \longmapsto \langle \varphi^2, \mathcal{M}^2, w \rangle$$

where $\langle \varphi, \mathcal{M}, w \rangle$ is an instance of $\text{IPC}^\rightarrow\text{-KMc}$. Claim 12 shows that $\mathcal{M}, w \models \varphi$ if and only if $\mathcal{M}^2, w \models \varphi^2$. It follows directly from the construction that this is a logspace reduction. □